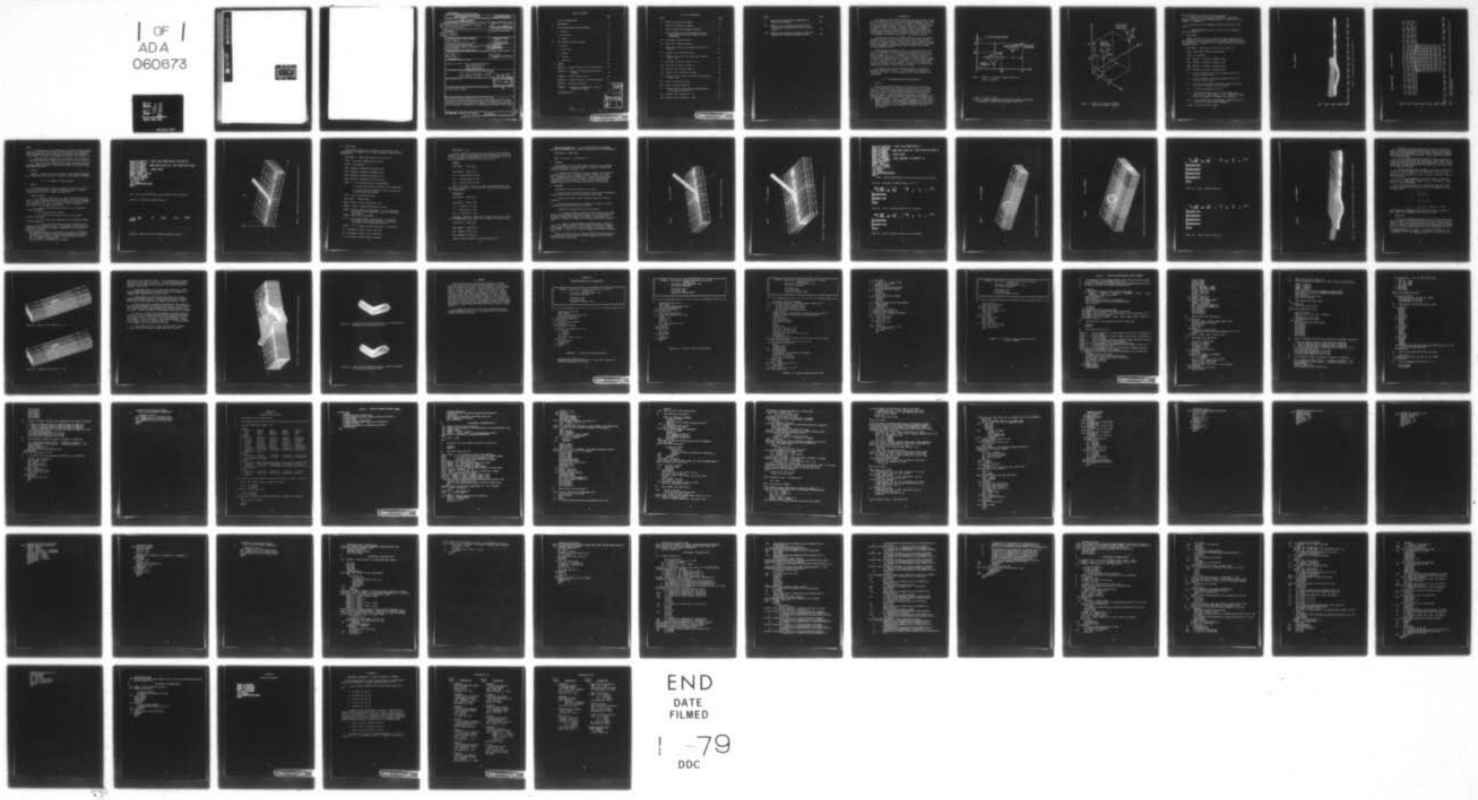
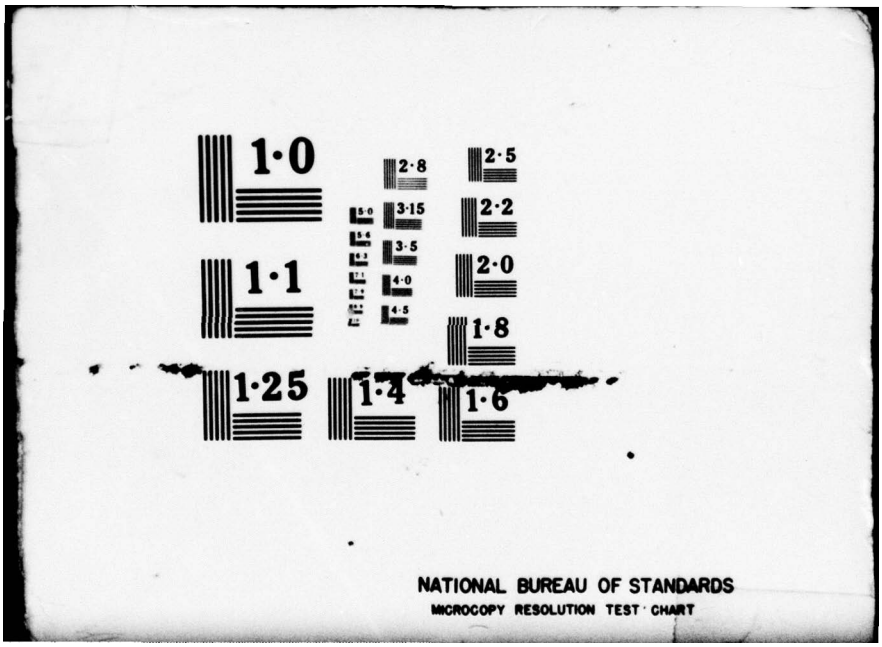
AD-A060 673 ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERD--ETC F/G 9/2
DEFORMATION PLOTTING PROGRAMS FOR THE EPIC FINITE ELEMENT CODES--ETC(U)
AUG 78 B E RINGERS

UNCLASSIFIED ARBRL-TR-02096 SBIE-AD-E430 113 NL

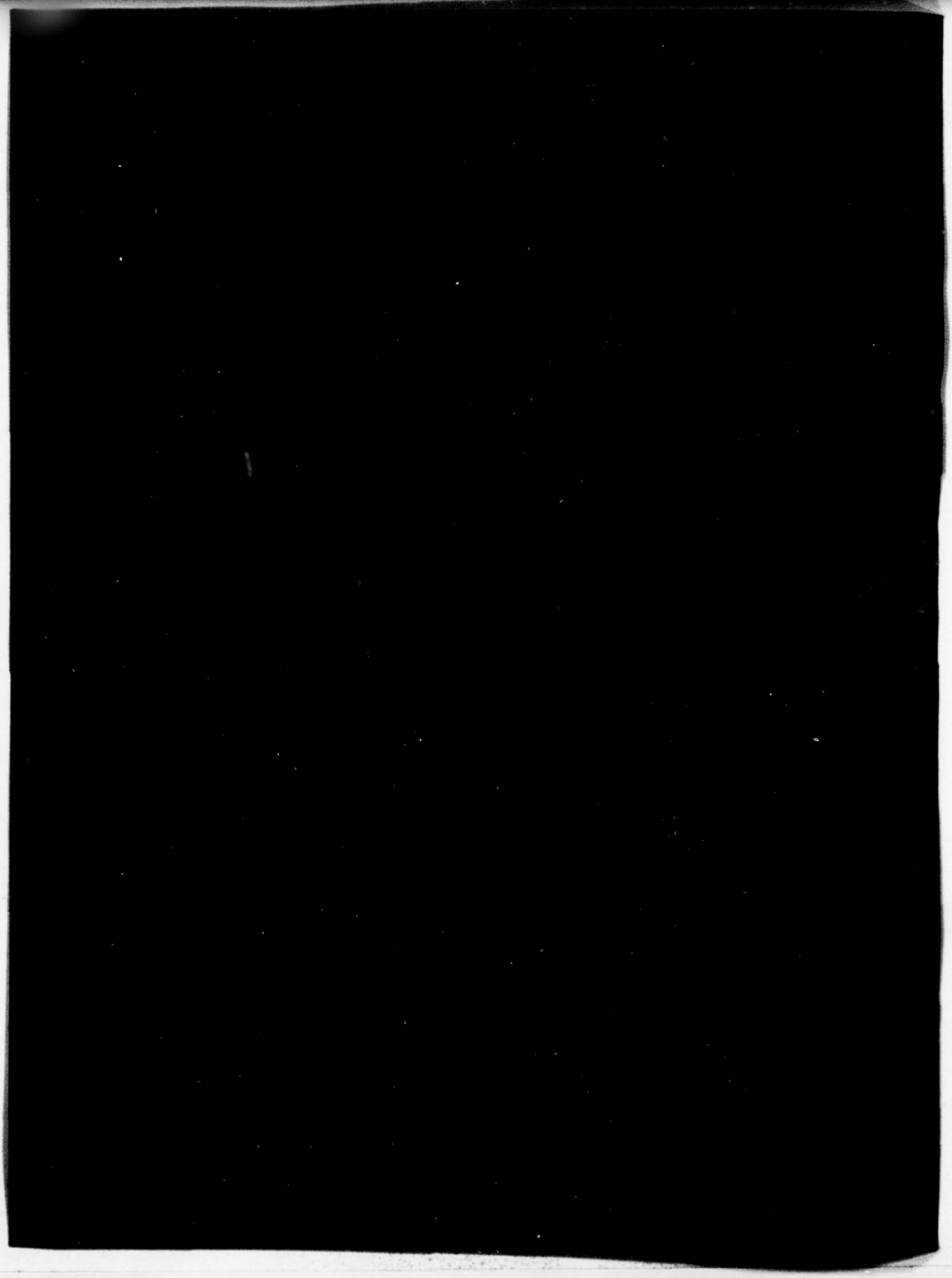
1 OF 1
AD A
060673



END
DATE
FILMED
1 -79
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

ARBRE



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) TECHNICAL REPORT ARBRL-TR-02096	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Deformation Plotting Programs For The Epic Finite Element Codes	5. TYPE OF REPORT & PERIOD COVERED (9) Final rept.; 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) B.E. Ringers	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: DRDAR-BLT Aberdeen Proving Ground, MD 21005	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) RDT&E 1L162618AH80
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research and Development Command US Army Ballistic Research Laboratory ATTN: DRDAR-BL Aberdeen Proving Ground, MD 21005	12. REPORT DATE (11) AUGUST 1978 / 13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 73 p.	15. SECURITY CLASS. (of this report) Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. (18) SBIE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) (19) AD-E430 113	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Plotting, Finite Element	

DDC RECEIVED NOV 1 1978 B

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (hmn)

This report describes deformation plotting routines written to handle output generated by two and three dimensional Epic finite element codes. The output contains the nodes which define each triangular or tetrahedral element and the nodal coordinates. Documentation is included detailing usage of these programs.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

393471

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

This report describes plotting routines which were written to handle data generated by two finite element computer programs, Epic-2[1] and Epic-3[2], and create cross-sectional or isometric deformation plots of the impact situation as predicted by the programs on a Calcomp Plotter or Tektronix Display Terminal. Geometry, in the Epic programs, is modeled as a series of triangular elements (Epic-2) and tetrahedral elements (Epic-3) defined by three and four nodes respectively. The displacements which may occur as the impact progresses are reflected in the changes in the nodal coordinates. (See Figures 1 and 2).

The nodes which define each element and their present coordinates constitute the plot files created at desired intervals during the running of the Epic Programs. Although the plotting programs described herein were written to handle Epic plot files specifically, they could readily be adapted to interface with other programs which generate the same geometrical data.

These plotting programs were written in standard Fortran for the Univac 1108 Computer but could easily be utilized on most other computers of equal size. It is assumed that the potential user of these plotting programs has access to either (A) a Tektronix Display Terminal which communicates with the computer on which the Epic program is run and the Tektronix Plot - 10 software is resident (Note changes to this in Appendix A for this program.) or, (B) a Calcomp Plotter wherein the standard Calcomp software is resident on the computer and automatically creates a file to drive the plotter.

The original cross-sectional plotting program was supplied by Honeywell with the Epic-3 code. It was modified by the author as discussed in Section II and is included in this report primarily for completeness.

II. THE CROSS-SECTIONAL PLOTTING PROGRAM

A. Options.

This plotting program handles plot files created by Epic-2 or Epic-3 and generates a cross-sectional deformation plot of an impact situation. All of the nodal coordinate connections are plotted for Epic-2 data whereas only those nodal coordinate connections between nodes for which Y = 0 are plotted for Epic-3 data. The original plotting program, provided by Honeywell, generated a plot tape for the Calcomp plotter. This version of the program (listed in Appendix B)

1. Johnson, Gordon R., "Epic-2 A Computer Program For Elastic-Plastic Impact Calculations in 2 Dimensions," Honeywell, Inc., Dec. 1977.
2. Johnson, Gordon R., "Epic-3 A Computer Program For Elastic-Plastic Impact Calculations in 3 Dimensions," BRL CR 343, July 1977. (AD #A043281)

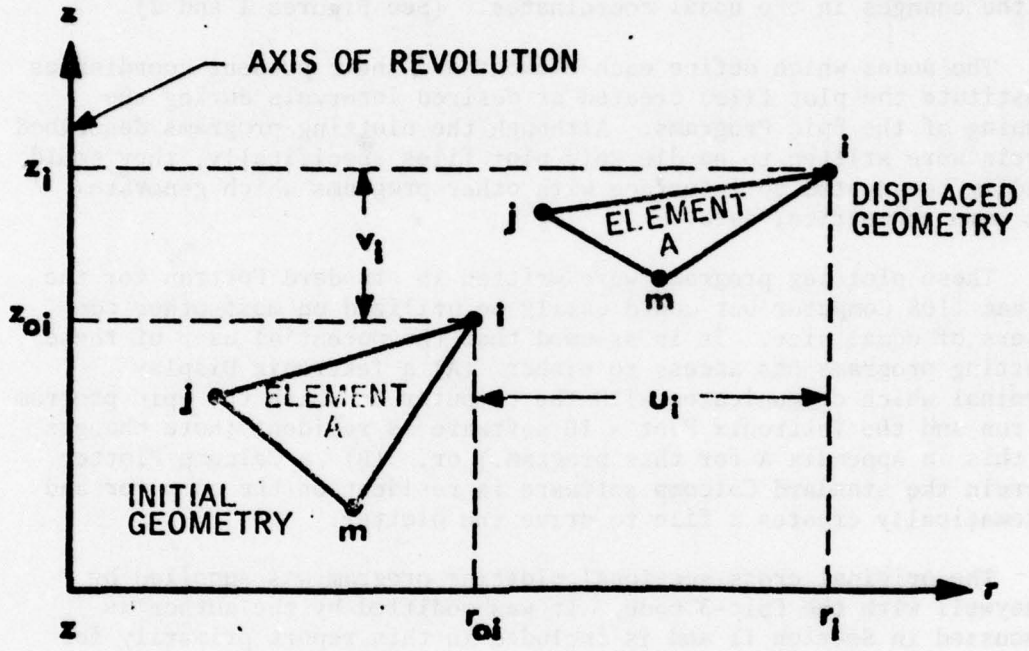


Figure 1. Geometry in the Epic-2 Program (Courtesy of Gordon R. Johnson[3])

3. "Epic-2 A Computer Program for Elastic-Plastic Impact Calculations in 2 Dimensions", Interim Technical Report, March 1977.

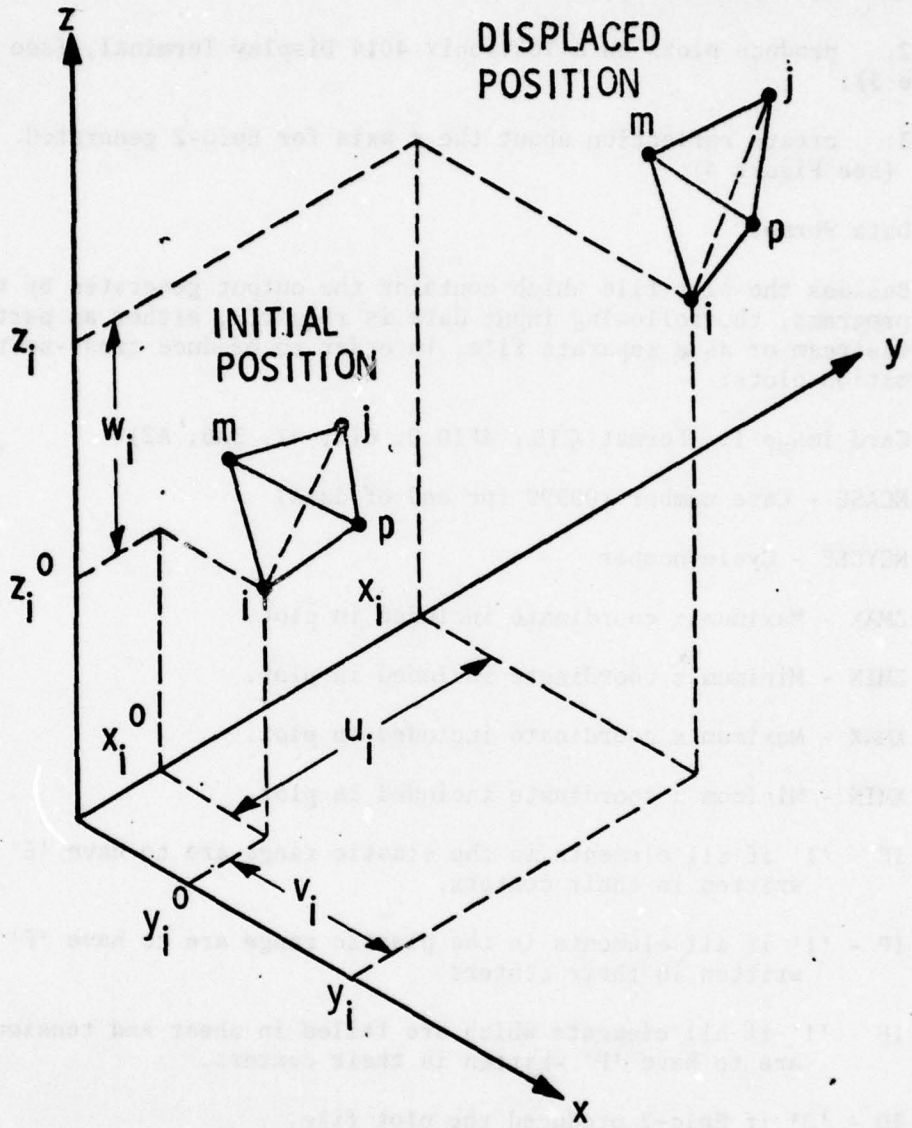


Figure 2. Geometry in the Epic-3 Program.
(Courtesy of Gordon R. Johnson[2])

has been modified to provide the following options:

1. handle output from the Epic-2 code also. (The GPLOT subroutine in Epic-2 was accordingly modified to interface with this program. See Appendix C);

2. produce plots on a Tektronix 4014 Display Terminal, (see Figure 3);

3. create reflection about the z axis for Epic-2 generated data, (see Figure 4).

B. Data Format.

Besides the plot file which contains the output generated by the Epic programs, the following input data is required, either as part of the runstream or as a separate file, in order to produce cross-sectional deformation plots:

Card image 1: Format (2I5 , 4F10.0, 6I1, 4X, 3A6, A2)

NCASE - Case number (09999 for end of data)

NCYCLE - Cycle number

ZMAX - Maximum z coordinate included in plot.

ZMIN - Minimum z coordinate included in plot.

XMAX - Maximum x coordinate included in plot.

XMIN - Minimum x coordinate included in plot.

IE - '1' if all elements in the elastic range are to have 'E' written in their centers.

IP - '1' if all elements in the plastic range are to have 'P' written in their centers.

IF - '1' if all elements which are failed in shear and tension are to have 'F' written in their centers.

ID - '2' if Epic-2 produced the plot file.

IR - '1' if reflection about z-axis is to be plotted also (Epic-2 only). (See Figure 4.) Note: XMIN will probably need to be changed to accommodate the reflection.

ICAL - '1' if Calcomp tape to be generated. Otherwise plot will be output to Tek 4014 screen immediately.

TITLE - Title to be written on plot.

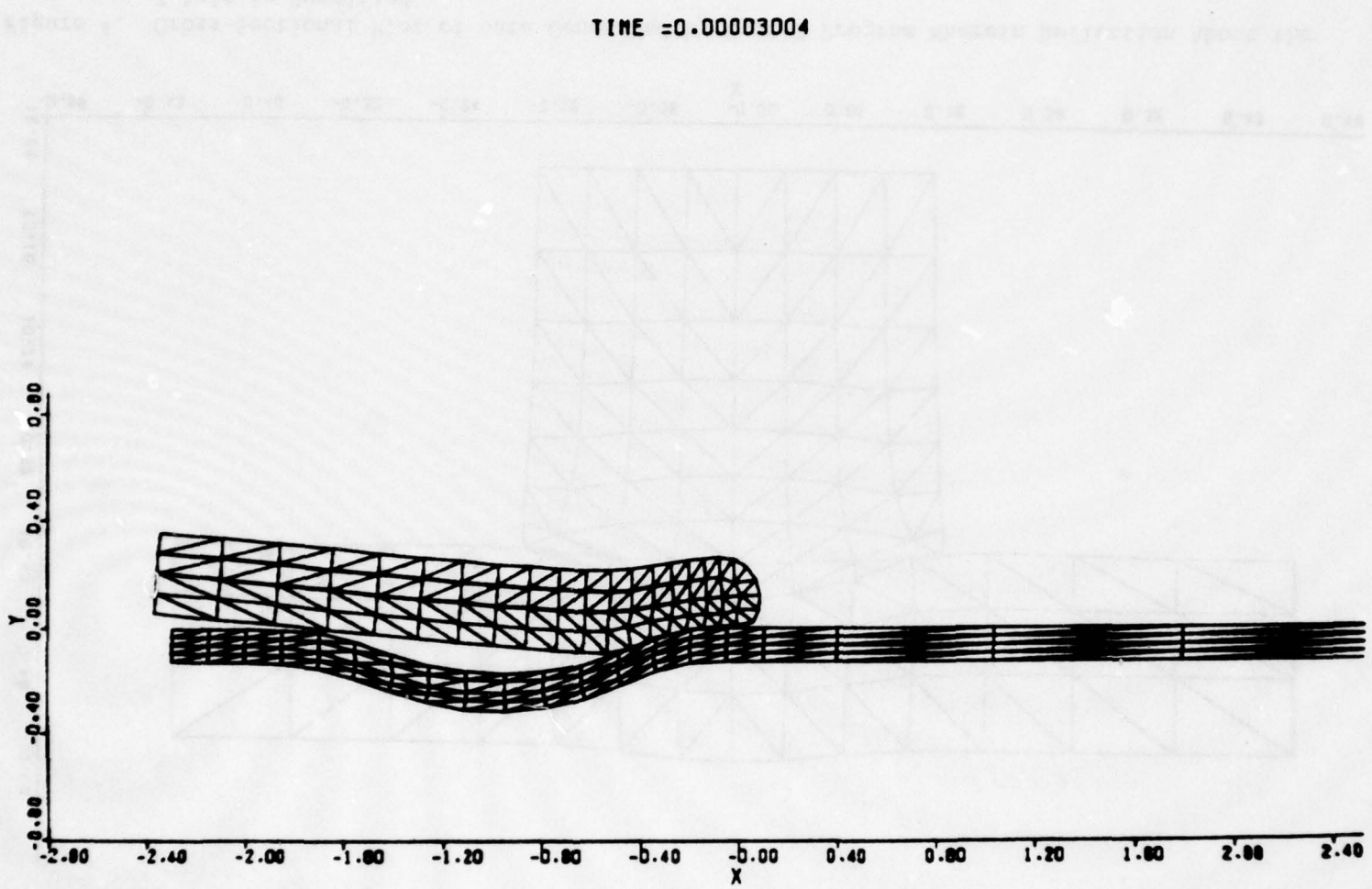


Figure 3. Cross-Sectional Plot of Impact Situation

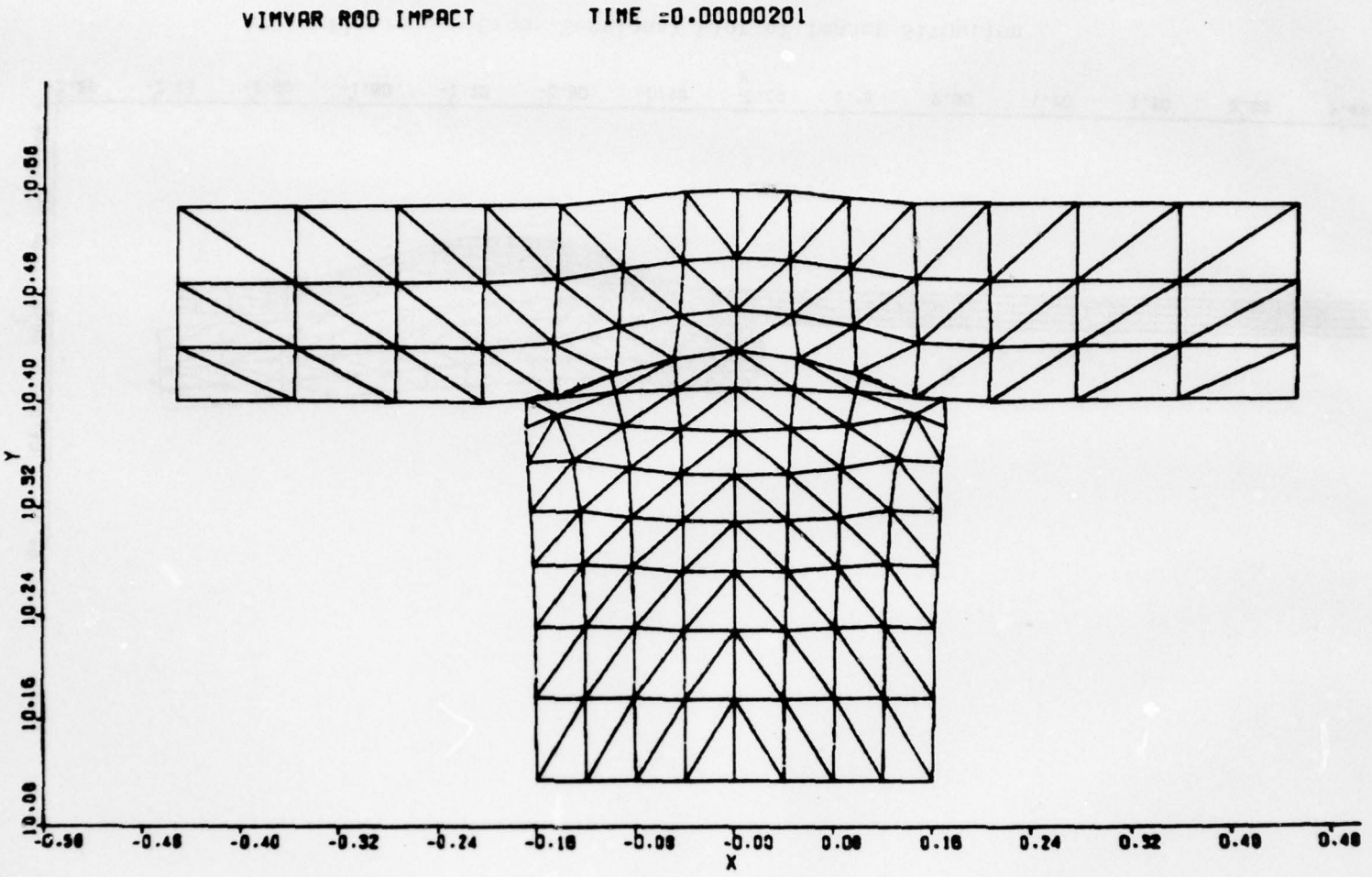


Figure 4. Cross-Sectional Plot of Data Generated by Epic-2 Program Wherein Reflection About the Z-Axis is Specified.

NOTE:

1. If the plots are to be generated on a Tektronix Display Terminal, each cycle specified correctly will be plotted and a copy automatically made on the hard copy unit (assuming one is connected to the terminal). Press RETURN, the screen will be erased and the next cycle plotted, etc.

2. You may specify as many cycles as desired as long as they are in ascending order by cycle number and the cycles exist on the same file.

3. If you do not get anything plotted, check your output file (6). You have probably specified a case and cycle number which are different from those on the file.

C. Example.

Figure 5A is the runstream for the Univac 1108 Computer and Figure 5B, the input data which was used to generate the plot shown in Figure 3.

III. THE ISOMETRIC PLOTTING PROGRAM

A. Options.

This plotting program (listed in Appendix D) handles output from Epic-3 only and generates an isometric deformation plot of an impact situation with the use of a hidden line algorithm.[4]

Options include:

1. creating a reflection of the nodal coordinates about the x-axis so that a complete projectile and target may be viewed, assuming that the geometry specified is for half the impact situation (the nodes are restrained in the y direction) and every nodal y coordinate > 0 (see Figure 6 for coordinate system used);

2. scaling, translating and rotating* the scene (specified in input parameters);

3. the following possible outputs:

a. a plot tape for future plotting on the Calcomp plotter (batch mode),

b. the scene plotted immediately on the Tektronix screen (interactive mode),

c. a plot tape (batch mode) for future plotting on the Tektronix screen (interactive mode). (See Appendix E for the retrieval runstream and Appendix A for changes to Tektronix Plot-10 routines to handle this situation).

---

4. The algorithm and basis for the hidden line subroutine (PLOT3D) is described in "Algorithm 483 Masked Three Dimensional Plot Program With Rotations," Steven L. Watkins, Communications of the ACM, Vol. 17, Number 9, September 1974.

* See Section IIIC for limitation on rotations.

```
@ASG,AX PRPLOT.  PLOT FILE FROM EPIC-2 OR EPIC-3
@USE 3,PRPLOT.
@ASG,AX RRDUM.  CASE AND CYCLE NO. FOR FILES ON FILE3
@USE 6,RRDUM.
@ASG,AX R2DATA.  INPUT DATA
@USE 7,R2DATA.
@PREP RPLOT.,RLIB.
@MAP ,TPF$.ABS
IN RPLOT.2DPLOT
IN RPLOT.TOP
LIB RLIB.
LIB SEAPPD*TEKLIB2.
@XQT
```

NOTE: RLIB and SEAPPD*TEKLIB2 contain Tektronix Plot-10 routines.

Figure 5A. Runstream to Generate Figure 3.

```
  114  380          .8       -0.8          2.8       -2.52
09999
```

Figure 5B. Input Data (in File R2DATA) to Generate Figure 3.

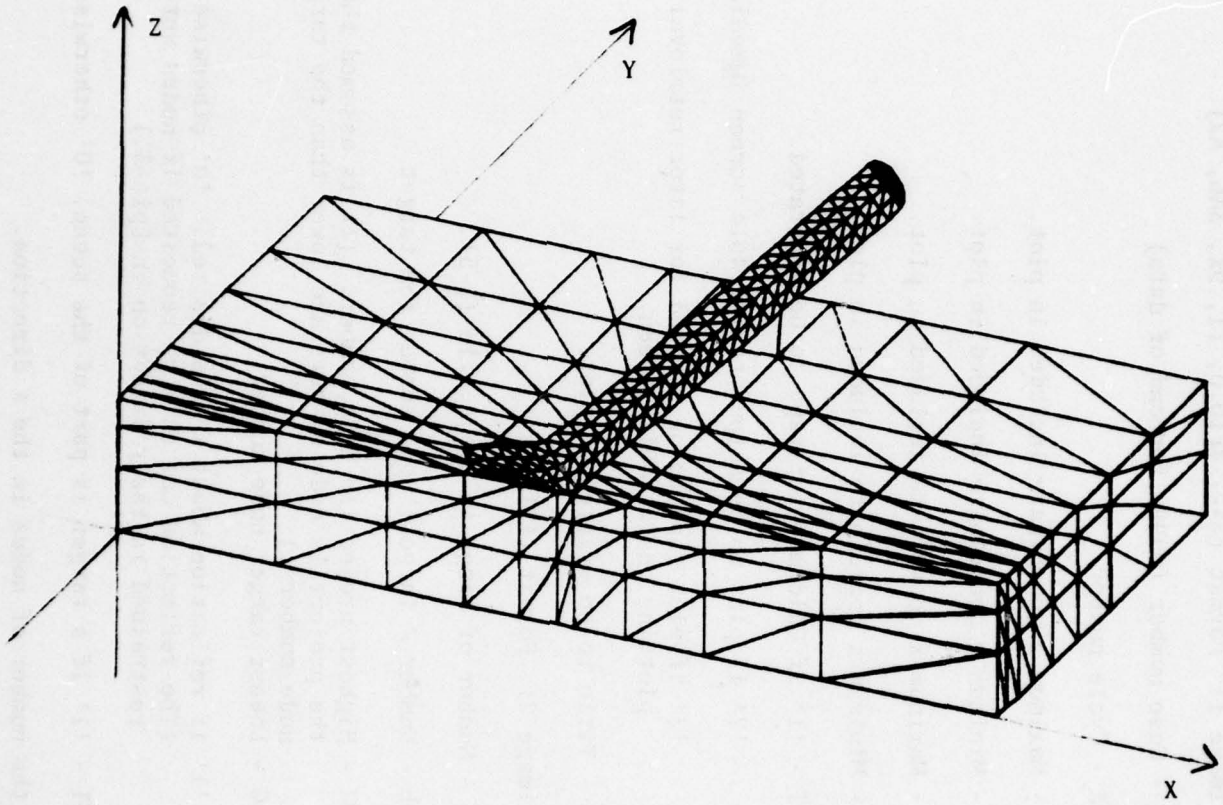


Figure 6. Right-handed Cartesian Coordinate System Used in Epic-3

B. Data Format.

The following input data is required, either as part of the runstream or as a separate file, in order to produce isometric deformation plots:

Card image 1: Format (2I5, 4F10.0, I1, 9X, 3A6, A2)

NCASE - Case number (09999 for end of data)

NCYCLE - Cycle number

ZMAX - Maximum z coordinate included in plot.

ZMIN - Minimum z coordinate included in plot.

XMAX - Maximum x coordinate included in plot.

XMIN - Minimum x coordinate included in plot.

IOUTPT - '1' if Calcomp plot tape to be generated.

'2' if plot to be output to Tek 4014 screen immediately.

'3' if plot tape to be created for later retrieval and plotting on Tek 4014 screen.

TITLE - Title to be written on plot.

Card image 2: Format (1315)

NRING - Number of rings in projectile (2-5).

NONODE - Number $\geq$ largest node number for target.

LNPROJ - Highest projectile node number. (It is assumed that the projectile node numbers are lower than the target node numbers.)

ISTARG - Lowest target node number.

IR - '1' if reflection about x axis desired. '0' otherwise. (The reflection can only be requested if nodes were restrained in the y direction in Epic-3.)

ITARGT - '1' if a target is part of the scene. '0' otherwise.

NX - the number of nodes in the x direction.

NY - the number of nodes in the y direction.

NZ - the number of nodes in the z direction.

Card image 3: etc:

Scaling, translation and rotation are specified in the following formats. The order in which they are placed should be the reverse of the order in which the transformations are to be performed, (i.e., the first transformation should be the last set of data).

Scaling:

Card image a: Format (I1)
'1'

Card image b: Format (3I4)

IS1 - x scaling factor (x 100)

IS2 - y scaling factor (x 100)

IS3 - z scaling factor (x 100)

Note: If scaling is done for at least one dimension it must be done for all three (e.g., if you want the z dimension doubled specify '1', '010001000200').

Translation:

Card image a: Format (I1)
'2'

Card image b: Format (3I4)

IT1 - x translation (x 100)

IT2 - y translation (x 100)

IT3 - z translation (x 100)

Rotation: Rotation is handled by rotating one axis into another (e.g., to rotate positively about y axis, rotate x axis into z axis)

Card image a: Format (I1)
'3'

Card image b: Format (3I4)

IR1 - number of first axis

IR2 - number of second axis

ITHETA - angle of rotation (in whole degrees only).

End of Transformations: If any transformations are performed they must be followed by this image indicating the end of transformations.

Card image a: Format (I1)
'4'

NOTE: See notes 1 - 3 in Section II.

C. Examples.

Plots generated for two target impact situations, one involving a long rod penetrator, the other a sphere, and their runstreams and input data are shown in Figures 7-10.

It is usually more impressive to see an isometric plot instead of a cross-sectional plot of an impact situation. However, in cases where a projectile causes deformation of a target but does not maintain continuous contact with the target, it may be more meaningful to generate a cross-sectional plot rather than an isometric plot. (Compare Figures 3 and 11 for which the same plot file was used.)

D. Procedure.

The procedure within the program is as follows:

1. Read the nodal coordinates representing the nodal displacements of the projectile and target at a given cycle (time step).

2. Create reflection of these coordinates about the x-axis if requested.

3. Perform transformations on coordinates according to scaling, translation and rotation parameters specified.

4. Perform the following steps for the projectile:

a. The four triangles, inherently created when the four nodal points of each tetrahedral element are joined, are checked against all the other triangles which make up the figure. If duplicated, the triangle is an interior triangle and is removed from further consideration.

b. Check all nodes remaining (defining exterior triangles) against all exterior triangles to see if they are hidden by a triangle, (see Appendix F for details of this procedure). If a node is hidden, any triangle, of which it is a vertex, is removed from further consideration.

Almost all the lines are now eliminated which are hidden from view for an individual object, that is, without regard to the relationship between projectile and target.

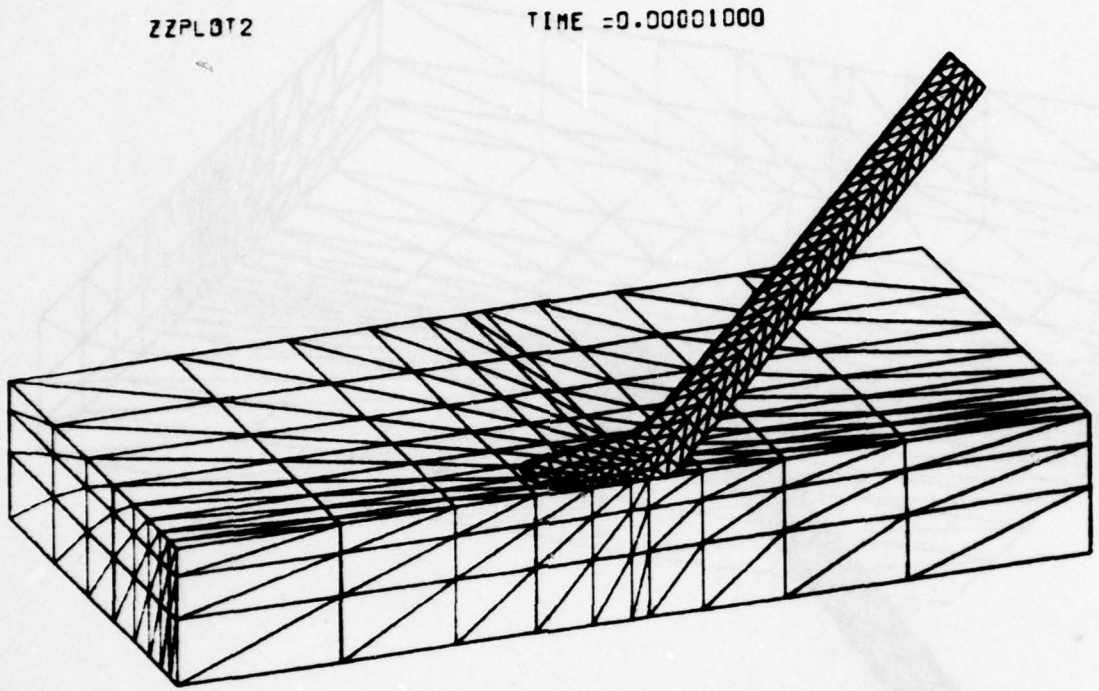


Figure 7A. Isometric Plot of Rod and Target

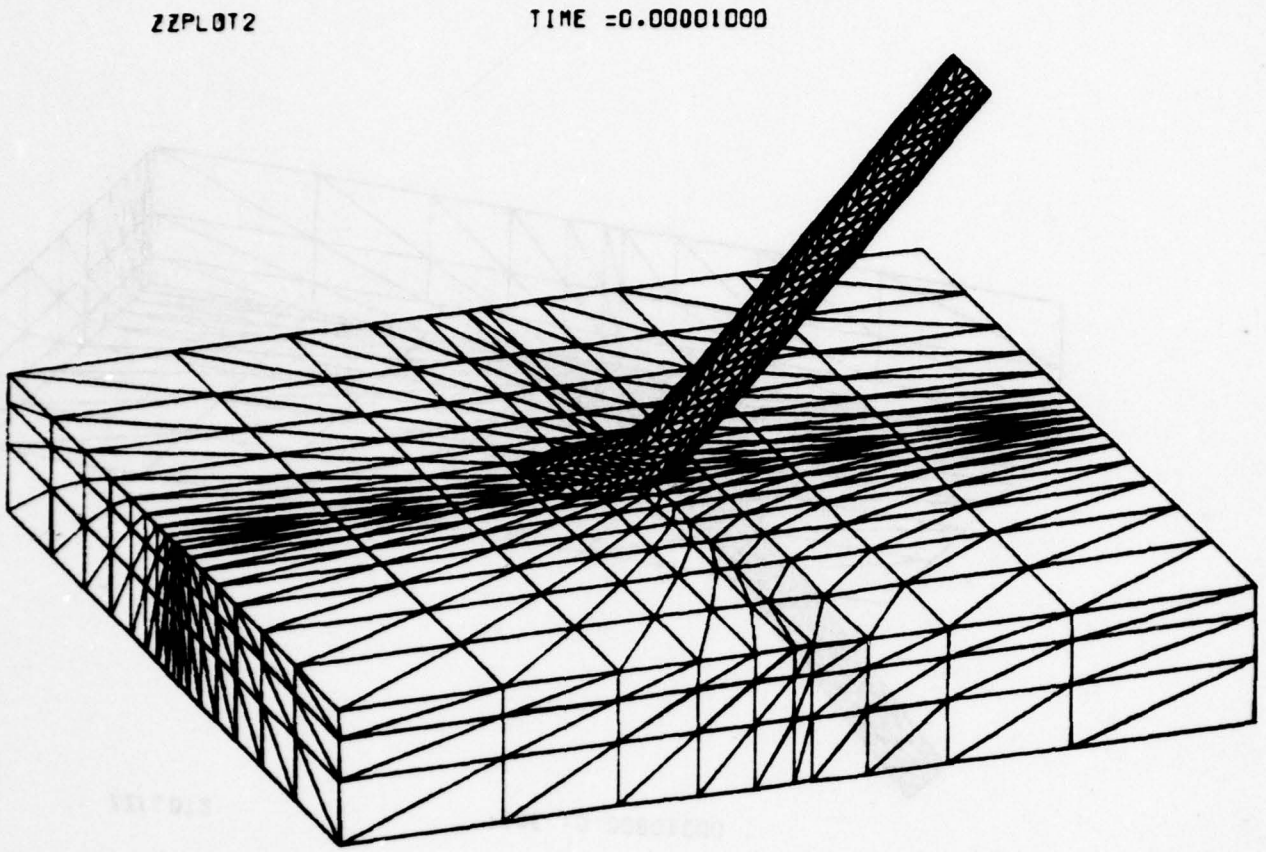


Figure 7B. Isometric Plot of Rod and Target With Reflection About X Axis

```
@ASG,AX ZZPLOT2.  PLOT FILE FROM EPIC-3
@USE 3,ZZPLOT2.
@ASG,AX RRDUM.  CASE AND CYCLE NO. FOR FILES ON FILE 3
@USE 6,RRDUM.
@ASG,AX RRDATA.  INPUT DATA
@USE 7,RRDATA.
@ASG,AX RDUM22.  FILE CREATED IF IOUTPT = 3
@USE 22,RDUM22.
@PREP RPLOT.
@MAP ,TPF$.ABS
IN RPLOT.3DPLOT
LIB RPLOT.
LIB RLIB.
LIB SEAPPD*TEKLIB2.
@XQT
```

(NOTE: RLIB and SEAPPD*TEKLIK2 contain Tektronix Plot-10 Routines)

Figure 8A. Runstream to Generate Figures 7A and 7B.

```
  602 1083        1.4       -.26          1.        -1.2
    2 1000  465  501          1   11     9     4
3
000200030020
3
00020001-020
4
09999
```

Figure 8B. Input to Generate Figure 7A (in file RRDATA)

```
  602 1083        1.4       -.26          1.        -1.2
    2 1000  465  501     1    1   11     9     4
3
000200030020
3
00020001-020
4
09999
```

Figure 8C. Input to Generate Figure 7B (in file RRDATA).

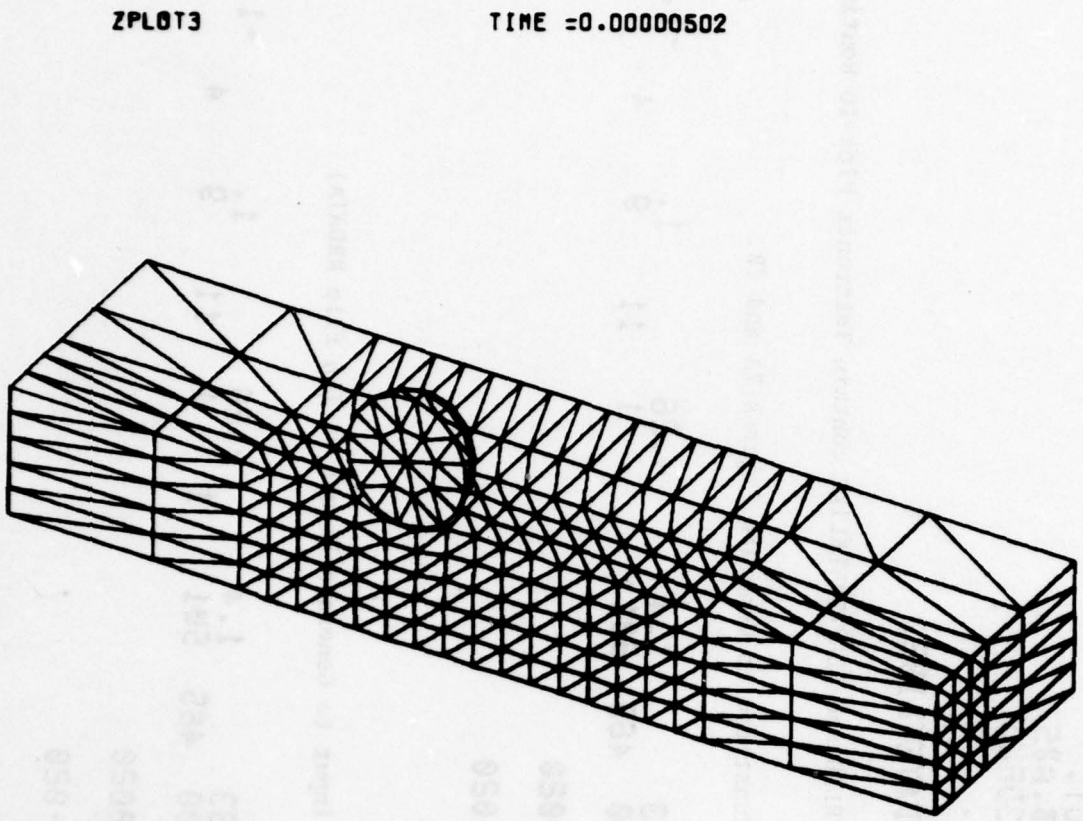


Figure 9A. Isometric Plot of Sphere and Target

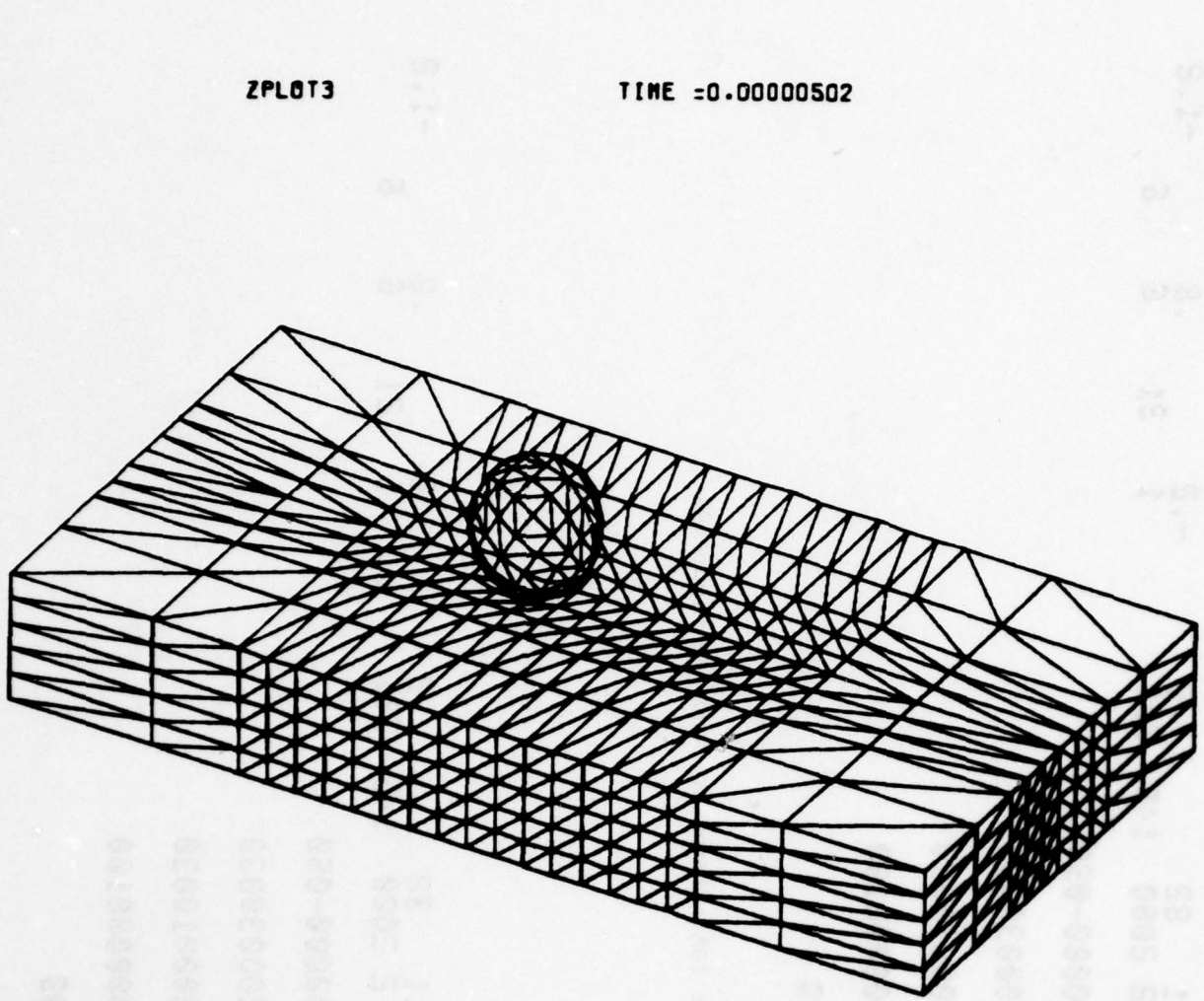


Figure 9B. Isometric Plot of Sphere and Target With Reflection About X Axis

```
  711    82          .2          -.2          .8          -1.2
    2 2000  145  301             1    21     6     6
2
00000000-050
3
000200030030
3
000200010030
2
000000000100
4
09999
```

Figure 10A. Input to Generate Figure 9A.

```
  711    82          .2          -.2          .8          -1.2
    2 2000  145  301    1        1    21     6     6
2
00000000-050
3
000200030030
3
000200010030
2
000000000100
4
09999
```

Figure 10B. Input to Generate Figure 9B.

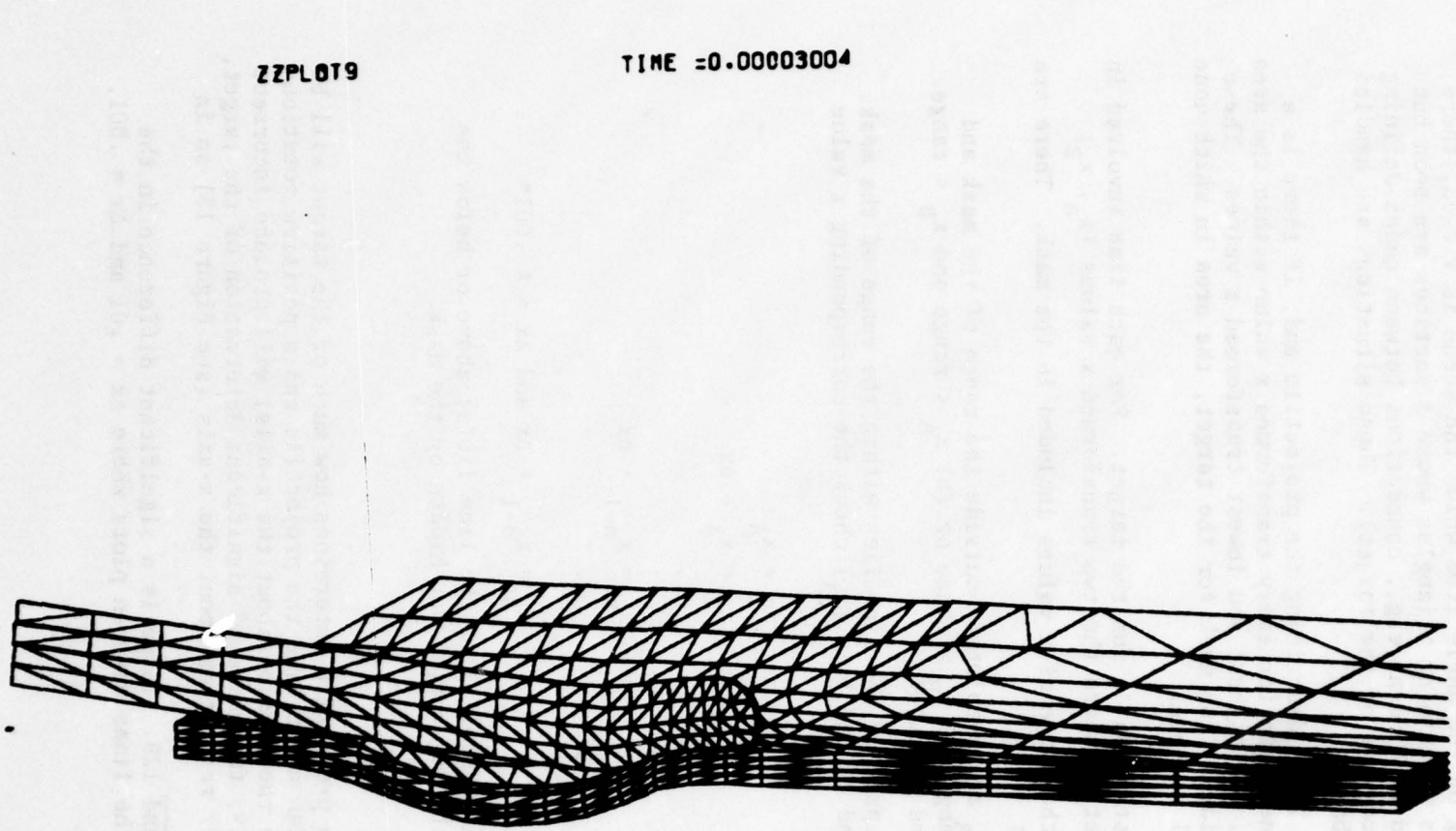


Figure 11. Isometric Plot of Impact Situation Wherein There is not Continuous Contact

c. Depending on the angle from which the scene is viewed, there remain a few lines describing triangles whose 3 vertices are seen but whose lines should be hidden (e.g., connections between nodes defining triangles at the corners of the target). These situations are handled on an individual basis.

d. Plot the lines describing the projectile and, if there is a target in the scene, note for every transformed x value within the area of the projectile, the highest and lowest transformed z values. These values form the hidden line mask for the target, the area in which none of the target will be seen.

5. Perform steps 4a-4c for the target. For each line involved in plotting the target, compare the two transformed x values (x_A, x_B) defining it with the range of x values included in the mask. There are two possibilities:

i. x_A and x_B are completely outside the range of the mask and either (a) x_A > range and x_B > range or (b) x_A < range and x_B < range. The line is plotted normally.

ii. at least part of the line lies within the range of the mask. Starting at one end of the line (x_A) check the corresponding z value for each x where

$$x_1 = x_A$$
$$x_2 = x_A + \Delta x$$
$$\vdots$$
$$x_n = x_{n-1} + \Delta x$$
$$\vdots$$
$$x_B = x_{B-1} + \Delta x \text{ and } \Delta x = \pm .001^*$$

to determine which segment(s) of the line lie(s) above or below the mask. Then plot the segment(s) not hidden by the mask.

E. Limitations.

1. Since the projectile determines how much of the target will be seen, the target can never hide the projectile and a positive rotation from the z-axis to the y-axis (about the x-axis) will produce incorrect results. Similarly, for cases of significant deformation of the target, one may not specify reflection about the x-axis (see Figure 13) so in

*See Figures 12A and 12B. There is a significant difference in the straightness of the lines between plots where $\Delta x = .01$ and $\Delta x = .001$.

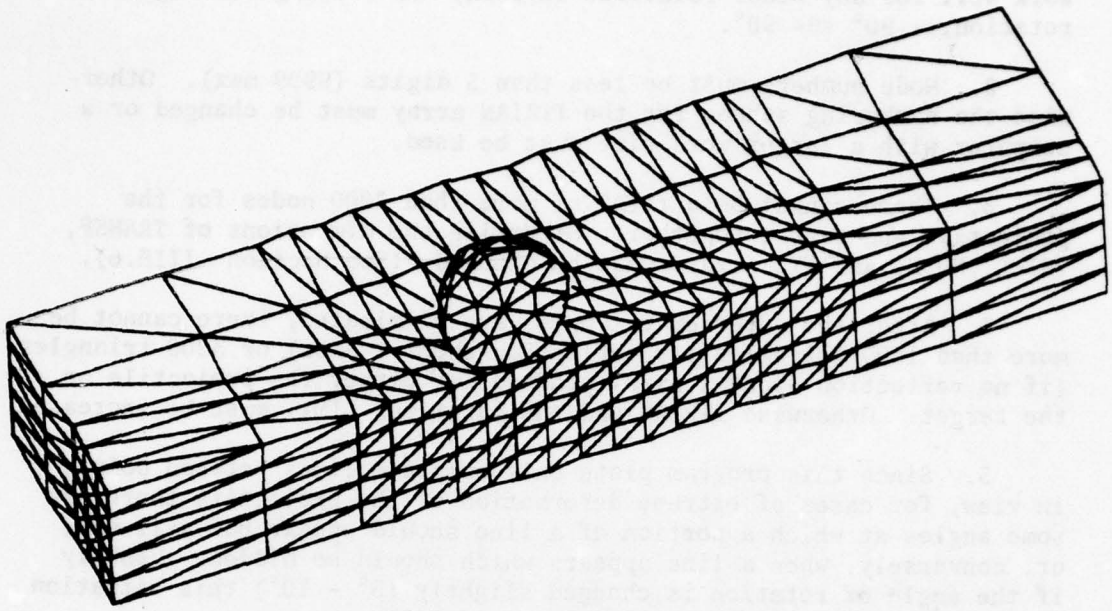


Figure 12A. Isometric Plot in Which $\Delta X = .01$

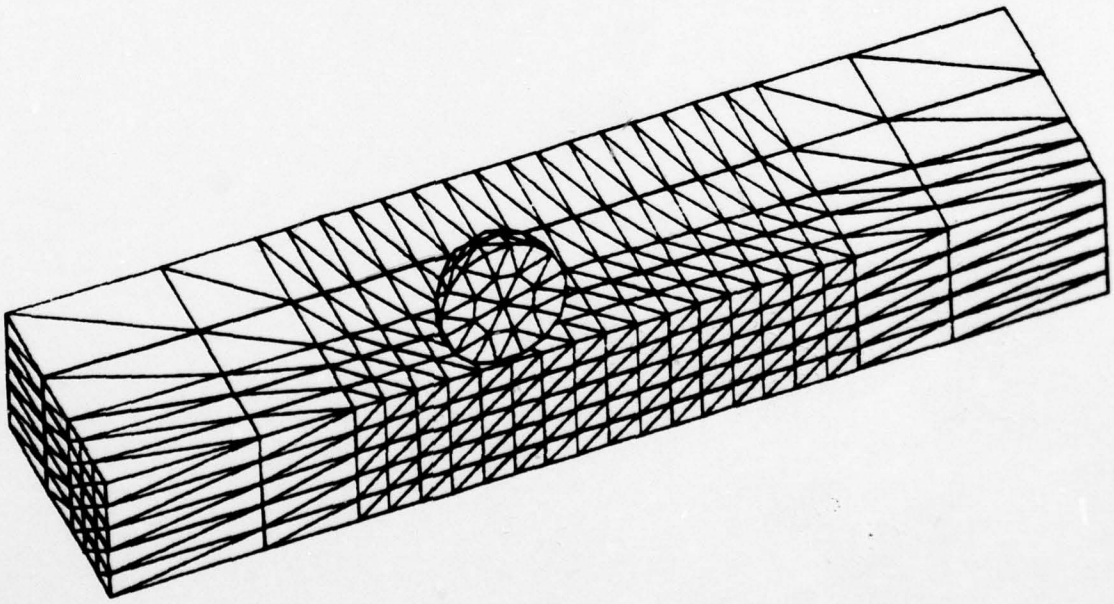


Figure 12B. Isometric Plot in Whic $\Delta X = .001$

Section IIIB, Card image 2 IR must = 0. This problem does not occur when there is no target in the scene. This plotting program should work well for any other rotations wherein, for θ being the angle of rotation, $-90° < \theta < 90°$.

2. Node numbers must be less than 5 digits (9999 max). Otherwise the numbering scheme for the ITRIAN array must be changed or a computer with a larger word size must be used.

3. There cannot be a total of more than 2000 nodes for the projectile and target together. Otherwise the dimensions of TRANSF, INTEL, NODE, and XYZ arrays must be changed, (See Section IIIE.6).

4. After the exterior triangles are eliminated, there cannot be more than 1600 triangles (if reflection about x axis) or 3200 triangles (if no reflection about x axis) plotted for either the projectile or the target. Otherwise ITRIAN and the parameter LIMIT must be increased.

5. Since this program plots only complete lines between points in view, for cases of extreme deformation of the projectile there are some angles at which a portion of a line should appear but will not, or, conversely, when a line appears which should be hidden. Usually if the angle of rotation is changed slightly (5° - 10°) this situation will no longer occur (See Figures 14A and 14B).

6. This program just fits a Univac 1108-65K memory computer so if any arrays are enlarged, extended memory must be used.

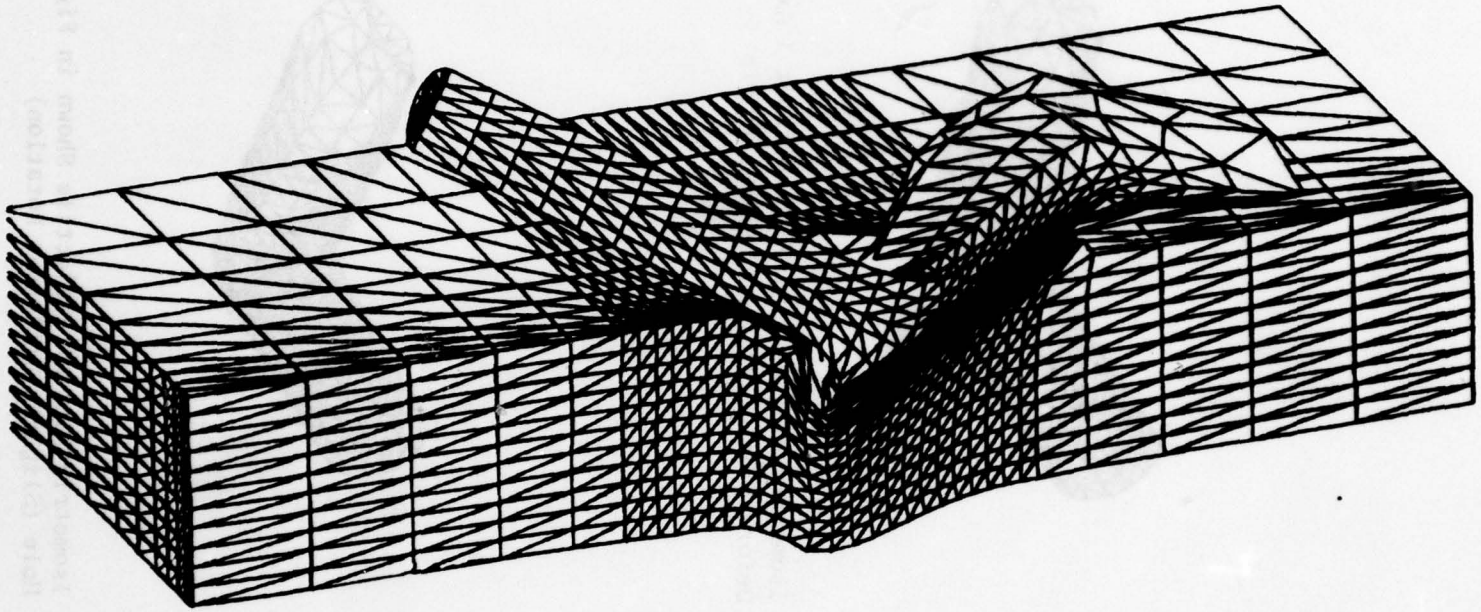


Figure 13. Isometric Plot Showing Severe Deformation of Projectile and Target

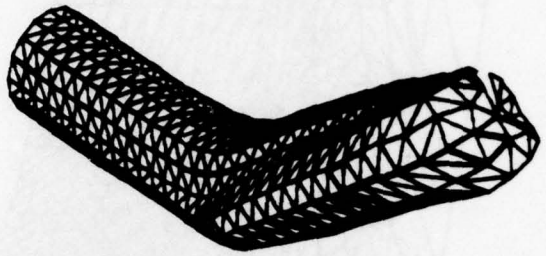


Figure 14A. Isometric Plot With Reflection About X Axis Showing Severe Deformation of Projectile With a Hole

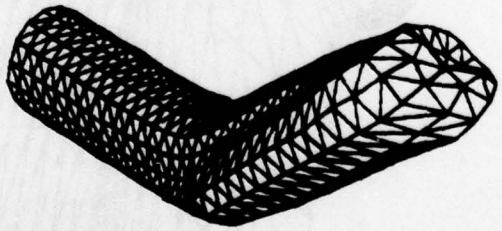


Figure 14B. Isometric Plot of Projectile Shown in Figure 14A Without Hole (Slightly Different Rotation)

IV. SUMMARY

These plotting routines were developed primarily to enable relatively quick and easy analysis of the deformation history of an impact as predicted by the Epic codes by providing a visual presentation of the nodal displacements. The key to the success of the isometric plotting program is, of course, the hidden line routine. Without it, there would only be a jumbled pile of lines. Ideally, one would employ a more sophisticated hidden line algorithm which would handle every possible situation and angle of view in creating isometric plots but that would require a great deal of computer time and would be extremely expensive. The techniques used in this program were geared to the specific scene arrangement we wanted for the impact simulations we handle and proved highly successful in terms of speed and cost.

It is hoped that other Epic users can profitably utilize the programs documented here to aid in the analysis of the deformations of their particular situations.

APPENDIX A*

REVISED TEKTRONIX PLOT 10 SUBROUTINES

```
C*************************************************************************
C*      PRODUCT  062-1526-02  PREVIEW ROUTINES FOR CALCOMP               *
C*                            RELEASE 1.1                                *
C*            C  COPYRIGHT 1973 TEKTRONIX, INC.                          *
C*               ALL RIGHTS RESERVED                                     *
C*                                                                       *
C*               TEKTRONIX, INC.                                         *
C*               P. O. BOX 500                                           *
C*               BEAVERTON, OREGON 97005                                 *
C*                                                                       *
C*************************************************************************
C
C-------------ROUTINE--PIN----------------------TEKTRONIX, INC.--------
C
      SUBROUTINE PIN(X,Y,N)
      REAL A(100)
      DATA IFIRST,NBMAX/989893,100/
      IF(IFIRST)100,300,100
C* FIRST PASS---OPEN FILE
  100 IFIRST = 0
C     CALL IFILE(22,'CTMP1')
C *FILL BUFFER
  200 READ(22,END=400)A
      NB = 0
  300 IF(NB .GT. NBMAX-3)GO TO 200
      X = A(NB+1)
      Y = A(NB+2)
      N = A(NB+3)
      NB = NB + 3
      IF(N .NE. 999)GO TO 500
  400 N = 999
      IFIRST=989893
C     ENDFILE 22
  500 RETURN
      END
```

Appendix A-1. Listing of Tektronix Routine PIN

*Permission was granted by Tektronix, Inc., May 1978 to document the following subroutines in this report.

```
C*****************************************************************************
C*      PRODUCT  062-1526-02  PREVIEW ROUTINES FOR CALCOMP                   *
C*                            RELEASE 1.1                                    *
C*            C  COPYRIGHT 1973 TEKTRONIX, INC.                              *
C*               ALL RIGHTS RESERVED                                         *
C*                                                                           *
C*               TEKTRONIX, INC.                                             *
C*               P. O. BOX 500                                               *
C*               BEAVERTON, OREGON 97005                                     *
C*                                                                           *
C*****************************************************************************
C
C-------------ROUTINE--POUT----------------------TEKTRONIX, INC.--------
C
      SUBROUTINE POUT(X,Y,N)
      REAL A(100)
      DATA IFIRST,NBMAX/989893,100/
      IF(IFIRST)100,200,100
C*FIRST PASS---OPEN FILE
  100 IFIRST = 0
C     CALL OFILE(22,'CTMP1')
  200 IF(NB .GE. NBMAX-3)GO TO 300
      A(NB+1) = X
      A(NB+2) = Y
      A(NB+3) = N
      NB = NB + 3
      IF(N .EQ. 999)GO TO 400
      GO TO 500
C *DUMP BUFFER
  300 WRITE(22)A
      NB = 0
      GO TO 200
  400 WRITE(22)A
      NB = 0
C     ENDFILE22
  500 RETURN
      END
```

Appendix A-2. Listing of Tektronix Routine POUT

```
C*************************************************************************
C*      PRODUCT  062-1526-02  PREVIEW ROUTINES FOR CALCOMP               *
C*                            RELEASE 1.1                                *
C*            C  COPYRIGHT 1973 TEKTRONIX, INC.                          *
C*               ALL RIGHTS RESERVED                                     *
C*                                                                       *
C*               TEKTRONIX, INC.                                         *
C*               P. O. BOX 500                                           *
C*               BEAVERTON, OREGON 97005                                 *
C*                                                                       *
C*************************************************************************
C
C----------SUBROUTINE--PLOT----------------------TEKTRONIX, INC.--------
C
      SUBROUTINE PLOT(XIN,YIN,NPEN)
C * THIS ROUTINE FOLLOWS THE NORMAL CALCOMP CONVENTION OF
C *       2= PEN DOWN OR VISIBLE VECTOR
C *       3= PEN UP OR NONVISIBLE VECTOR
C *      -2= NEW ORIGIN VISIBLE VECTOR
C *      -3= NEW ORIGIN NONVISIBLE VECTOR
      LOGICAL SKIP
      COMMON /CLCOMP/ XSTART,YSTART,XFSET,YFSET,XACUM,YACUM,
     + SKIP,NSKIP,IOPT,XLEN,YLEN,XFAC,YFAC,FAC,NHARD
      LOGICAL SAVE
      LOGICAL ANOPLT
      DATA SAVE/.TRUE./
      DATA ANOPLT/.FALSE./
      X=XIN
      Y=YIN
      N=IABS(NPEN)
      IF(N .EQ. 995) GO TO 995
      IF(N .EQ. 997)GO TO 997
      IF(N .EQ. 999) GO TO 999
      IF(SKIP .AND. NPEN .GT. 0) GO TO 900
      NSKIP=NSKIP-1
      SKIP=NSKIP .GT. 0
C IF IT IS DESIRED TO APPLY OFFSETS AND SCALING TO ALL VECTORS
C REMOVE NEXT STATEMENT
      IF(N .NE. 12 .AND. N .NE. 13) GO TO 001
C  APPLY OFFSETS AND FACTORS
      N=N-10
      X=X*XFAC+XFSET
      Y=Y*YFAC+YFSET
001   X=X*FAC+XACUM
      Y=Y*FAC+YACUM
      IF(SAVE)CALL POUT(X-XACUM,Y-YACUM,NPEN)
      IF(ANOPLT)GO TO 100
      IF(N .EQ. 3) GO TO 003
      IF(N .EQ. 2) GO TO 002
C  ERROR IN CALL
      GO TO 900
C  DRAW IF NPEN=2, -2, 12, -12
002   CALL DRAWA(X,Y)
      GO TO 100
C  MOVE IF NPEN=3, -3, 13, -13
003   CALL MOVEA(X,Y)
```

Appendix A-3. Listing of Tektronix Routine PLOT

```
C  FINAL CHECK
100    IF(NPEN .GT. 0) GO TO 900
       IF(SKIP) GO TO 900
C  EXERCISE OPTION
       GO TO(101,102,103,105),IOPT
C  1=CHANGE ORIGIN AND PROCEDE
101    XACUM=X
       YACUM=Y
       GO TO 900
C  2=CHANGE ORIGIN AND INQUIRE
102    XACUM=X
       YACUM=Y
       GO TO 104
C  3=CHANGE TO USER ORIGIN AND INQUIRE
103    XACUM=XSTART
       YACUM=YSTART
104    CONTINUE
C  4=USER DEFINED OPTION
C  USER MAY INSERT CODE HERE
C  OPTION MUST BE ALTERED TO PERMIT 4
105    GO TO 900
C      CHANGE SAVE FLAG
 995    SAVE=.NOT.SAVE
       GO TO 900
  997 ANOPLT = .TRUE.
       GO TO 900
900    RETURN
 999    IF(SAVE)CALL POUT(X,Y,N)
       CALL FINITT(0,0)
       GO TO 900
       END
```

```
C*************************************************************************
C*      PRODUCT  062-1526-02  PREVIEW ROUTINES FOR CALCOMP               *
C*                            RELEASE 1.1                                *
C*              C  COPYRIGHT 1973 TEKTRONIX, INC.                        *
C*                 ALL RIGHTS RESERVED                                   *
C*                                                                       *
C*                 TEKTRONIX, INC.                                       *
C*                 P. O. BOX 500                                         *
C*                 BEAVERTON, OREGON 97005                               *
C*                                                                       *
C*************************************************************************
C
C-------------ROUTINE--MAINLINE RETRIEVAL SYSTEM-TEKTRONIX, INC.--------
C
      CALL INITT(30)
      CALL PLOT(0.,0.,995)
  100 CALL PLOTS(0,0,0)
  200 CALL PIN(X,Y,N)
      IF(N .EQ. 999)GO TO 999
      CALL PLOT(X,Y,N)
      GO TO 200
C* END OF PASS
  999 CALL TOUTPT(7)
      CALL TOUTPT(7)
      CALL HDCOPY
      CALL TINPUT(K)
C * A 'Q' WILL STOP THE JOB
      IF(K .NE. 81)GO TO 100
      CALL FINITT(0,0)
      END
```

Appendix A-4. Listing of Tektronix Mainline Retrieval Routine - RETREV

APPENDIX B. LISTING OF CROSS-SECTIONAL PLOTTING PROGRAM

```
C
C     THIS PROGRAM PLOTS DEFORMED GEOMETRY FROM EPIC-2 OR EPIC-3 OUTPUT.
C     IT PLOTS THE X-Z PLANE GEOMETRY AT Y = 0.
C   THE DATA IS OBTAINED FROM TAPE 3, WRITTEN IN SUBROUTINE GPLOT,EPIC-2
C    OR TAPE 13, WRITTEN IN SUBROUTINE GPLOT,EPIC-3
C
C
      DIMENSION
     1  NODE(4000),X(4000),Y(4000),Z(4000),IFIX(4000),
     *    SYM(3),     SYMB(3),    XSIZE(4),   ZSIZE(5),   XP(6),    ZP(6)
        DIMENSION TITLE(4),IBUF(1000)
      DATA XX,YY/1HX,1HY/
      DATA BLANK/6H      /
      DATA SYM(1),SYM(2),SYM(3)/ 1HE,1HP,1HF/
      DATA LIT1,LIT2,LIT3,LIT4,LIT5/49,50,51,52,53/
      DATA IFIRST/0/
C
      YTITLE=YY
      XTITLE=XX
  100 FORMAT(2I5,4F10.0,6I1,4X,3A6,A2)
  101 FORMAT (/////,15X,23HEND OF TAPE 3 PLOT DATA)
  102 FORMAT (1H1,5X,7HCASE  =,I5,/,5X,7HCYCLE =,I5,/,5X,7HIPLOT =,I5)
  103 FORMAT (4X,I5,3E15.6,5X,I5)
  104 FORMAT (1H1,////,9X,38HELE  NODE1  NODE2  NODE3  NODE4 ICHECK,///)
  105 FORMAT (5X,6I7)
C
C     INITIALIZE PLOT AND ESTABLISH ORIGIN AT LOWER LEFT
C
      REWIND 3
      REWIND 7
C
C     READ INPUT DATA FOR PLOT
C
C**** IE = 1 IF ALL ELEMENTS IN ELASTIC RANGE ARE TO HAVE 'E' WRITTEN IN
C****       THEIR CENTERS.
C**** IP = 1 IF ALL ELEMENTS IN PLASTIC RANGE ARE TO HAVE 'P' WRITTEN IN
C****       THEIR CENTERS.
C**** IF = 1 IF ALL ELEMENTS WHICH ARE FAILED IN SHEAR AND TENSION ARE TO
C****       HAVE 'F' WRITTEN IN THEIR CENTERS.
C**** ID = 2 IF 2D VERSION OF EPIC PRODUCED PLOT TAPE
C**** IR = 1 IF AXISYMMETRIC PROBLEM AND REFLECTION ABOUT Z AXIS IS TO
C****       PLOTTED ALSO AND ID = 2.
C**** ICAL = 1 IF CALCOMP PLOT TAPE TO BE CREATED. OTHERWISE PLOT WILL
C**** BE OUTPUT TO TEKTRONIX 4014 SCREEN.
  150 READ(7,100)NCASE,NCYCLE,ZMAX,ZMIN,XMAX,XMIN,IE,IP,IF,ID,IR,ICAL
     *,TITLE
      IF(NCASE.EQ.9999) GO TO 900
      IF(IFIRST .EQ. 0)CALL PLOTS(IBUF,1000,22)
      IF(ICAL .EQ. 0 .OR. IFIRST .EQ. 1) GO TO 153
      CALL PLOT(0.,-10.,-3)
      CALL PLOT(0.,1.25,-3)
      IFIRST = 1
  153 SYMB(1)=SYM(1)
      SYMB(2)=SYM(2)
      SYMB(3)=SYM(3)
```

```
      XSIZE(1)=XMIN
      XSIZE(2)=XMAX
      ZSIZE(1)=ZMIN
      ZSIZE(2)=ZMAX
      DELTX = ABS(XMAX - XMIN)
      DELTZ = ABS(ZMAX - ZMIN)
      IF(ICAL .EQ. 1)GO TO 154
      YRANGE = DELTZ/11.
      XRANGE = DELTX/14.
      GO TO 157
  154 YRANGE = DELTZ/8.5
      XRANGE = DELTX/11.
  157 USE = AMAX1(YRANGE,XRANGE)
      YRANGE =  DELTZ/USE
      XRANGE = DELTX/USE
      CALL SCALE(XSIZE,XRANGE,2,1)
      CALL SCALE(ZSIZE,YRANGE,2,1)
      XP(5)=XSIZE(3)
      XP(6)=XSIZE(4)
      ZP(5)=ZSIZE(3)
      ZP(6)=ZSIZE(4)
C
C     READ INITIAL DATA FROM TAPE 3
C
  155 CONTINUE
      READ (3) ICASE, ICYCLE, NNODE, NELE, TIME
      IF(ICYCLE.GE.0) GO TO 200
C     WRITE (6,101)
      GO TO 900
  200 IPLOT  = 1
      IF(NCASE .LT. 0)GO TO 201
      IF(ICYCLE.EQ.NCYCLE.AND.ICASE.EQ.NCASE) GO TO 201
      IPLOT  = 0
  201 WRITE (6,102) ICASE, ICYCLE, IPLOT
C
C     READ NODE DATA FROM TAPE 3
C
      IF(ID .NE. 2)GO TO 203
      DO 170 J = 1,NNODE
      READ(3)I,X(I),Z(I)
      X(I+2000) = -X(I)
      Z(I+2000) = Z(I)
C     WRITE(6,103)I,X(I),Z(I)
  170 CONTINUE
      GO TO 204
  203 READ (3) (NODE(I), I=1,NNODE)
      DO 202 J=1,NNODE
      I       = NODE(J)
      READ (3) X(I), Y(I), Z(I), IFIX(I)
      IF(IPLOT .NE. 1)GO TO 202
      IF(Y(I).NE.0.) GO TO 202
C     WRITE (6,103) I, X(I), Y(I), Z(I), IFIX(I)
  202 CONTINUE
  204 IF(IPLOT.NE.1) GO TO 300
C     WRITE (6,104)
```

```
C
C     SET UP PLOT AXES IF IPLOT = 1
C     DRAW 10.0 INCH AXIS SCALED FOR Z AXIS (X AXIS THE SAME SCALE)
C
      SUBZ3 = ZSIZE(3)
      SUBZ4 = ZSIZE(4)
      SUBX3 = XSIZE(3)
      SUBX4 = XSIZE(4)
      CALL AXIS(.5,.5,YTITLE,1,YRANGE,90.0,SUBZ3,SUBZ4)
      CALL AXIS(.5,.5,XTITLE,-1,XRANGE,0.0,SUBX3,SUBX4)
      CYCLE=NCYCLE
      CALL TOP (TIME,CYCLE,TITLE)
      CALL PLOT(.5,.5,-3)
C
C     READ ELEMENT DATA FROM TAPE 3
C
  300 CONTINUE
      IF(ID .NE. 2)GO TO 399
C
C****< TAPE CREATED BY EPIC-2
C
      DO 500 I=1,NELE
      READ (3) NEL, N1, N2, N3,  ICHECK,D1
      ISW = 0
      IF(IPLOT.NE.1) GO TO 500
      IF(ICHECK.EQ.3) GO TO 500
  350 CONTINUE
C     WRITE(6,105)NEL,N1,N2,N3,ICHECK
      XP(1)=X(N1)
      XP(2)=X(N2)
      XP(3)=X(N3)
      XP(4)=X(N1)
      ZP(1)=Z(N1)
      ZP(2)=Z(N2)
      ZP(3)=Z(N3)
      ZP(4)=Z(N1)
C
C     CHECK IF TRIANGLE VIOLATES PLOT BOUNDARIES (XMIN-XMAX, ZMIN-ZMAX)
C
      IF(X(N1).GT.XMAX.OR.X(N2).GT.XMAX.OR.X(N3).GT.XMAX.OR.
     *   Z(N1).GT.ZMAX.OR.Z(N2).GT.ZMAX.OR.Z(N3).GT.ZMAX.OR.
     *   X(N1).LT.XMIN.OR.X(N2).LT.XMIN.OR.X(N3).LT.XMIN.OR.
     *   Z(N1).LT.ZMIN.OR.Z(N2).LT.ZMIN.OR.Z(N3).LT.ZMIN) GO TO 370
      CALL LINE(XP,ZP,4,1,0,0)
      IF(IE.EQ.0.AND.ICHECK.EQ.0) GO TO 370
      IF(IP.EQ.0.AND.ICHECK.EQ.1) GO TO 370
      IF(IF.EQ.0.AND.ICHECK.EQ.2) GO TO 370
C
C     WRITE MATERIAL SYMBOL AT CENTER OF TRIANGLE IF SPECIFIED
C
      XS=(.333333*(XP(1)+XP(2)+XP(3)) - (XSIZE(3)))/XSIZE(4) - .02
      ZS=(.333333*(ZP(1)+ZP(2)+ZP(3)) - (ZSIZE(3)))/ZSIZE(4) - .04
      ICHEC=ICHECK+1
      SUBSYM = SYMB(ICHEC)
      CALL SYMBOL(XS,ZS,.07,SUBSYM,0.0,1)
```

```
  370 IF(ISW .EQ. 1 .OR. IR .NE. 1)GO TO 500
      ISW = 1
      N1 = N1 + 2000
      N2 = N2 + 2000
      N3 = N3 + 2000
      GO TO 350
  500 CONTINUE
      GO TO 600
C
C**** TAPE CREATED BY EPIC-3
C
  399 DO 400 I=1,NELE
      READ (3) NEL, N1, N2, N3, N4, ICHECK
      IF(IPLOT.NE.1) GO TO 400
      IF(ICHECK.EQ.3) GO TO 400
C
C     THREE POINTS (NP1,NP2,NP3) MAKE A PLANE
C
  410 CONTINUE
      NP1=N1
      NP2=N2
      NP3=N3
      IBACK=1
      GO TO 460
  420 CONTINUE
      NP1=N1
      NP2=N2
      NP3=N4
      IBACK=2
      GO TO 460
  430 CONTINUE
      NP1=N1
      NP2=N3
      NP3=N4
      IBACK=3
      GO TO 460
  440 CONTINUE
      NP1=N2
      NP2=N3
      NP3=N4
      IBACK=4
  460 IF(Y(NP1).EQ.0.AND.Y(NP2).EQ.0.AND.Y(NP3).EQ.0) GO TO 470
      GO TO (420,430,440,450),IBACK
  450 CONTINUE
C
C     NONE OF THE PLANES ARE ON THE Y=0.0 AXIS
C
      GO TO 400
C 470 WRITE (6,105) NEL, N1, N2, N3, N4, ICHECK
  470 CONTINUE
C
C     PLOT A TRIANGULAR TET FACE AT Y = 0.
C
      XP(1)=X(NP1)
      XP(2)=X(NP2)
```

```
      XP(3)=X(NP3)
      XP(4)=X(NP1)
      ZP(1)=Z(NP1)
      ZP(2)=Z(NP2)
      ZP(3)=Z(NP3)
      ZP(4)=Z(NP1)
C
C     CHECK IF TRIANGLE VIOLATES PLOT BOUNDARIES (XMIN-XMAX, ZMIN-ZMAX)
C
      IF(X(NP1).GT.XMAX.OR.X(NP2).GT.XMAX.OR.X(NP3).GT.XMAX.OR.
     *   Z(NP1).GT.ZMAX.OR.Z(NP2).GT.ZMAX.OR.Z(NP3).GT.ZMAX.OR.
     *   X(NP1).LT.XMIN.OR.X(NP2).LT.XMIN.OR.X(NP3).LT.XMIN.OR.
     *   Z(NP1).LT.ZMIN.OR.Z(NP2).LT.ZMIN.OR.Z(NP3).LT.ZMIN) GO TO 400
      CALL LINE(XP,ZP,4,1,0,0)
      IF(IE.EQ.0.AND.ICHECK.EQ.0) GO TO 400
      IF(IP.EQ.0.AND.ICHECK.EQ.1) GO TO 400
      IF(IF.EQ.0.AND.ICHECK.EQ.2) GO TO 400
C
C     WRITE MATERIAL SYMBOL AT CENTER OF TRIANGLE IF SPECIFIED
C
      XS=(.333333*(XP(1)+XP(2)+XP(3)) - (XSIZE(3)))/XSIZE(4) - .02
      ZS=(.333333*(ZP(1)+ZP(2)+ZP(3)) - (ZSIZE(3)))/ZSIZE(4) - .04
      ICHEC=ICHECK+1
      SUBSYM = SYMB(ICHEC)
      CALL SYMBOL(XS,ZS,.07,SUBSYM,0.0,1)
  400 CONTINUE
  600 IF(IPLOT.EQ.0) GO TO 155
C
C     GO TO END OF PLOT AND ESTABLISH NEW ORIGIN 2.0 INCHES OVER
C
      IF(ICAL .EQ. 1)GO TO 700
      CALL PLOT(0.,0.,999)
      CALL HDCOPY
      CALL TINPUT(NHDCPY)
      CALL PLOT(-.5,-.5,-3)
      GO TO 150
  700 OVER = OVER + 13.
      CALL PLOT(OVER,1.25,-3)
      CALL PLOT(-.5,-.5,-3)
      GO TO 150
  900 REWIND 3
      CALL PLOT(0.,0.,999)
      STOP
      END
```

```
      SUBROUTINE TOP(TIME,CYCLE,TITLE)
C        PLOT TITLE AND HEADER INFORMATION
C
         DIMENSION TITLE(3)
      CALL SYMBOL(2.5,8.,.14,TITLE,0.0,18)
      CALL SYMBOL(6.0,8.,.14,6HTIME =,0.0,6)
      CALL NUMBER(999.,999.,.14,TIME,0.0,8)
      RETURN
      END
```

APPENDIX C

REVISED GPLOT IN EPIC-2

```
      SUBROUTINE GPLOT (NCYCLE,TIME)
C
C     THIS ROUTINE WRITES DATA ON TAPE 3 FOR CALCOMP OR TEK 4014 PLOTS
C
C     REVISED VERSION SEPTEMBER, 1977
C
      COMMON
     *   NNODE,      NPNODE,     NELE,       NPELE,      ENERGY,
     *   DT1,        DTMAX,      DTMIN,      TMAX,       NDEP,
     *   PMAS,       TMAS,       NSLIDE,     NCASE,      SSF
      COMMON
     *   DEN(10),    SPH(10),    E(10),      V(10),      VIS(10),
     *   FY(10),     FU(10),     EU(10),     SBM(10),    RATE(10),
     *   CP1(10),    CP2(10),    CT1(10),    CT2(10),    EM(10),
     *   VE(10),     EBS(10),    C(10),      D(10),      S(10),
     *   G(10),      Q1(10),     Q2(10),     TEMP1(10),  EYY(10),
     *   EG(10),     ELAM(10),   B(10),      A(10),      DETVEL(10),
     *   EQST1(10),  EQST2(10),  EQST3(10),  EQST4(10),  EQST5(10),
     *   EQST6(10),  EQST7(10),  EQST8(10),  EQST9(10),  EQST10(10),
     *   GAMMA(10),  EE(10)
      COMMON
     *   R(1000),    Z(1000),    RI(1000),   ZI(1000),   RDOT(1000),
     *   ZDOT(1000), FR(1000),   FZ(1000),   NODE(1000), AMASS(1000),
     *   PMASS(1000),IFIX(1000)
      COMMON
     *   NEL(2000),  NODE1(2000),NODE2(2000),NODE3(2000),ICHECK(2000),
     *   VOL(2000),  DVOL(2000), AREA(2000), MAT(2000),  AI(2000),
     *   ES(2000),   EW(2000),   DAREA(2000),DDVOL(2000),TSTART(2000),
     *   BURNFR(2000)
      COMMON
     *   IKIND(5),   NMNODE(5),  NSNODE(5),  NSELE(5),   IEND1(5),
     *   IEND2(5),   IMN(5,50),  ISN(5,50),  ISE(5,50),  RM(50),
     *   ZM(50)
C
  115 FORMAT (/////,10X,5HCYCLE,I5,30H DATA SAVED ON TAPE 3 FOR PLOT)
C
      WRITE (3) NCASE, NCYCLE, NNODE, NELE,TIME
C
      DO 100 J=1,NNODE
      I       = NODE(J)
  100 WRITE (3) I, R(I), Z(I)
C
      DO 200 J=1,NELE
      I       = NEL(J)
  200 WRITE (3) I,NODE1(I),NODE2(I),NODE3(I), ICHECK(I), DDVOL(I)
C
      WRITE (6,115) NCYCLE
C
      RETURN
      END
```

APPENDIX D. LISTING OF ISOMETRIC PLOTTING PROGRAM

```
COMDIM* PROC
        COMMON
       *ALPHA,NPNTS,IMAT,ISTART,IEND
       *,SYM(3),SYMB(3),XSIZE(4),ZSIZE(5),XP(6),ZP(6),NP(4)
       *,NCYCLE,NCASE
       *,TRANSF(4000,4),INTEL(4000)
       *,NODLAY,NONODE,ISTARG,LNPROJ,IR,ITRIAN(3200,2)
       *,ITARGT,IN,LIMIT
       *,ARRAY(4,4),ARRINT(4,4),ARRCHG(4,4),IROTAT
       *,MASK(20000)
  END
```

```
      INCLUDE COMDIM,LIST
      DIMENSION TITLE(4),LAYNOD(5),NODE(2000),XYZ(4000,4)
     *,IBUF(1000)
      EQUIVALENCE (MASK,NODE),(MASK(2001),XYZ(1,1))
      DATA LAYNOD/6,15,24,37,50/
      DATA IFIRST/0/
C
C                          *****FORMAT STATEMENTS*****
C
  110 FORMAT (2I5,4F10.0,I1,9X,3A6,A2)
  120 FORMAT (1H1,5X,7HCASE  =,I5,/,5X,7HCYCLE =,I5,/,5X,7HIPLOT =,I5)
  130 FORMAT (16I5)
  140 FORMAT (' INDEX = ',I10)
  180 FORMAT (' ITRIAN NEEDS TO BE REDIMENSIONED LARGER')
C**** LIMIT IS DIMENSION OF ITRIAN***************
C****
      LIMIT = 3200
C****
C
C     INITIALIZE PLOT AND ESTABLISH ORIGIN AT LOWER LEFT
C
      REWIND 3
      REWIND 7
C
C     READ INPUT DATA FOR PLOT
C
C**** IOUTPT = 1 IF CALCOMP PLOT TAPE TO BE GENERATED.
C****        = 2 IF PLOT TO BE OUTPUT TO TEK 4014 SCREEN IMMED.
C****        = 3 IF PLOT TAPE TO BE GENERATED, LATER
C****          RETRIEVED AND PLOTTED ON TEK 4014 SCREEN.
C**** NRING = NO. RINGS/PROJECTILE
C**** NONODE > OR = HIGHEST NODE NUMBER FOR TARGET
C**** ISTARG = 1ST NODE NUMBER ON TARGET
C**** LNPROJ = HIGHEST PROJECTILE NODE NUMBER
C**** IR = 1 IF YOU WANT Y REFLECTION OF PROJECTILE AND TARGET
C****  CREATED. OTHERWISE IR = 0
C**** ITARGT = 1 IF THIS PLOT INCLUDES A TARGET(IE. HIDDEN
C****  LINE ALGORITHM MUST BE USED).
C**** NX = NUMBER OF TARGET ELEMENTS ALONG X AXIS
C**** NY = NUMBER OF TARGET ELEMENTS ALONG Y AXIS
C**** NZ = NUMBER OF TARGET ELEMENTS ALONG Z AXIS
  183 READ (7,110) NCASE,NCYCLE,ZMAX,ZMIN,XMAX,XMIN,IOUTPT,TITLE
      IF(NCASE .EQ. 9999)GO TO 610
      READ (7,130) NRING,NONODE,LNPROJ,ISTARG,IR,ITARGT,NX,NY
     *,NZ
C****INITIALIZE IN FOR CKTRI SUBROUTINE. IN = NO. TRIANGLES
C****  ALREADY CONSIDERED + 1
      IN = 1
C**** IMAT = 1 FOR PROJECTILE
C****      = 2 FOR TARGET
      IMAT = 1
C**** NODLAY = NUMBER NODES/LAYER OF PROJECTILE
      NODLAY = LAYNOD(NRING)
      DO 187 I = 1,4000
      INTEL(I) = 0
```

```
      DO 186 L = 1,4
  186 TRANSF(I,L) = 0.
  187 CONTINUE
      IMOD=LAYNOD(NRING - 1)
      JMOD=IMOD +NONODE
      KMOD=NODLAY+NONODE
C**** THE FOLLOWING ASSUMES 2 LE NRING LE 5
      ICK1 = (NRING - 2) * 2 + 6
      ICK2 = ICK1 + 2
C**** SET IND1 AND IND2 FOR CKPTS SO THAT NODES 1 THRU LNPROJ ARE
C****   CHECKED AGAINST EXTERIOR TRIANGLES OF PROJECTILE
      IND1 = 1
      IND2 = LNPROJ
      IRELEM=2*NONODE
      IREF = NONODE * 10000 + NONODE
      IF (IR.EQ.0) IRELEM=NONODE
  185    DO 190 I=1,IRELEM
         XYZ(I,4)=1
  190    CONTINUE
C
      SYMB(1)=SYM(1)
      SYMB(2)=SYM(2)
      SYMB(3)=SYM(3)
C**** BASED ON 11 X 14 TEKTRONIX 4014 SCREEN DETERMINE MAXIMUM
C**** SIZE PLOT CAN BE TO FIT ON SCREEN
      XSIZE(1)=XMIN
      XSIZE(2)=XMAX
      ZSIZE(1)=ZMIN
      ZSIZE(2)=ZMAX
      DELTX=ABS(XMAX-XMIN)
      DELTZ=ABS(ZMAX-ZMIN)
      IF(IOUTPT .EQ. 1)GO TO 195
      YRANGE=DELTZ/11.
      XRANGE=DELTX/14.
      GO TO 197
  195 YRANGE = DELTZ/8.5
      XRANGE = DELTX/11.
  197 USE=AMAX1(YRANGE,XRANGE)
      YRANGE=DELTZ/USE
      XRANGE=DELTX/USE
      CALL SCALE (XSIZE,XRANGE,2,1)
      CALL SCALE (ZSIZE,YRANGE,2,1)
      XP(5)=XSIZE(3)
      XP(6)=XSIZE(4)
      ZP(5)=ZSIZE(3)
      ZP(6)=ZSIZE(4)
C
C     READ INITIAL DATA FROM TAPE 3
C
  200 READ (3) ICASE,ICYCLE,NNODE,NELE,TIME
      IF (ICYCLE.GE.0) GO TO 210
C     WRITE (6,101)
      STOP
  210 IPLOT=1
      IF (ICYCLE.EQ.NCYCLE.AND.ICASE.EQ.NCASE) GO TO 220
```

```
      IPLOT=0
  220 WRITE (6,120) ICASE,ICYCLE,IPLOT
C
C     READ NODE DATA FROM TAPE 3
C
      READ (3) (NODE(I),I=1,NNODE)
         DO 230 J=1,NNODE
         I=NODE(J)
         READ (3) XYZ(I,1),XYZ(I,2),XYZ(I,3),IFIX
  230    CONTINUE
      IF (IPLOT.EQ.0) GO TO 300
      IF (IR.EQ.0) GO TO 250
C**** CREATE MIRROR IMAGE OF PENETRATOR
         DO 240 J=1,NNODE
         I=NODE(J)
         XYZ(I+NONODE,1)=XYZ(I,1)
         XYZ(I+NONODE,2)=-XYZ(I,2)
         XYZ(I+NONODE,3)=XYZ(I,3)
  240    CONTINUE
C**** READ IN SCALING,TRANSLATION,ROTATIONAL PARAMETERS
  250 CALL TRANS
C**** TRANSFORM COORDINATES
C
         DO 280 I=1,IRELEM
            DO 270 L=1,4
               DO 260 K=1,4
               TRANSF(I,L)=XYZ(I,K)*ARRINT(K,L)+TRANSF(I,L)
  260          CONTINUE
  270       CONTINUE
  280    CONTINUE
  283 ALPHA = 3.14159/2.
C**** INITIALIZE MASK SO THAT TARGET WILL NOT BE DRAWN WHERE IT
C**** COINCIDES WITH PROJECTILE.
C
  285    DO 290 K=1,20000
         MASK(K)=-10000
  290    CONTINUE
      CYCLE=NCYCLE
      IF(IOUTPT .NE. 1 .OR. IFIRST .EQ. 0)
     *CALL PLOTS(IBUF,1000,22)
      IF(IOUTPT .NE. 1 .OR. IFIRST .EQ. 1)GO TO 295
      IFIRST = 1
      CALL PLOT(0.,-10.,-3)
      CALL PLOT(0.,1.25,-3)
  295 IF(IOUTPT .EQ. 3)CALL PLOT(0.,0.,997)
      CALL TOP (TIME,CYCLE,TITLE)
C
C     READ ELEMENT DATA FROM TAPE 3
C
  300    DO 600 IA=1,NELE
         READ (3) NEL,N1,N2,N3,N4,ICHECK
         IF (IPLOT.EQ.0) GO TO 600
C**** CHECK TO SEE IF FIRST TARGET NODES READ YET, IF SO
C****  PROCESS PROJECTILE NODES
      IF(N1 .GT. LNPROJ .AND. IMAT .EQ. 1)GO TO 315
```

```
C****CHECK 4 TRIANGLES DEFINED BY 4 POINTS READ
C****IGNORE INTERIOR ONES.
  305    CALL CKTRI (N1,N2,N3,N4)
      IF(IA .NE. NELE)GO TO 600
C****WHEN LAST SET OF POINTS READ OR, IF TARGET,
C****FIRST SET OF TARGET POINTS READ, COME HERE.
  315 CONTINUE
      IF (IR.NE.1) GO TO 330
C****PACK ITRIAN SO THERE'LL BE ROOM FOR REFLECTIVE TRIANGLES.
      CALL PACK (INDEX)
      WRITE (6,140) INDEX
      IF ((INDEX+INDEX).GT.LIMIT) GO TO 520
C****IF Y REFLECTION OF PROJECTILE AND TARGET DESIRED CREATE
C****REFLECTIVE TRIANGLES(EXTERIOR ONLY).
         DO 320 I=1,INDEX
         ITRIAN(I+INDEX,1)=ITRIAN(I,1)+IREF
      ITRIAN(I + INDEX,2) = ITRIAN(I,2) + NONODE
 320     CONTINUE
C****CHECK ALL POINTS AGAINST EXTERIOR TRIANGLES TO SEE WHICH
C****POINTS ARE HIDDEN AND SHOULD NOT BE CONNECTED.
 330  CALL CKPTS(IND1,IND2)
      NPNTS = 4
         DO 490 JJ=1,LIMIT
         IF (ITRIAN(JJ,1).EQ.0) GO TO 490
C****RETRIEVE NODAL POINTS FROM ITRIAN
      M1 = ITRIAN(JJ,1) / 10000
      M2 = ITRIAN(JJ,1) - (M1 * 10000)
      M3 = ITRIAN(JJ,2)
C****IF ONE OF THE THREE POINTS DEFINING A TRIANGLE IS HIDDEN,
C****NO PART OF TRIANGLE IS PLOTTED.
      IF(INTEL(M1) .EQ. 1 .OR. INTEL(M2) .EQ. 1 .OR.
     *   INTEL(M3) .EQ. 1)GO TO 490
C****SINCE A TRIANGLE IS PLOTTED WHEN ALL 3 POINTS DEFINING IT ARE SEEN,
C****THERE ARE SOME ANGLES AT WHICH AN ENTIRE PROJECTILE
C****IS PLOTTED WHEREIN SOME LINES AT THE FAR END OF THE
C****PROJECTILE ARE UNDESIRABLE.
C****
      IF(IMAT .EQ. 2)GO TO 340
      IF(IR .EQ. 1)GO TO 407
C
C**** PROJECTILE ONLY - NO REFLECTION
C
      GO TO 410
C
C**** HALF AND FULL TARGET
C
C**** THERE ARE ALSO SOME ANGLES AT WHICH THE TARGET IS
C**** PLOTTED WHEREIN SOME LINES NEAR CORNERS ARE UNDESIRABLE
  340 KT1 = M1 - ISTARG + 1
      KT2 = M2 - ISTARG + 1
      KT3 = M3 - ISTARG + 1
      NXNY = NX * NY
      NXNYM = NXNY + ISTARG - NX
      NXNYMR = NXNYM + NONODE
C**** DO NOT PLOT TRIANGLE JOINING UPPER LEFT BACK CORNER
```

```
C**** OF TARGET TO POINT BELOW IT AND TO ITS RIGHT
      IF(IABS(M1-M2) .EQ. 1 .AND. IABS(M1-M3) .EQ. NXNY
     * .AND. MOD(KT1,NX) .EQ. 1 .AND. MOD(KT3,NX) .EQ. 1
     * .AND. M1 .EQ. NXNYM)
     *GO TO 490
      IF(IR .EQ. 0)GO TO 405
C
C**** FULL TARGET
C
C**** REFLECTIVE TRIANGLES CREATED FOR 1ST NX NODES OF TARGET
C****DO NOT SHOW UP AS DUPLICATE INTERIOR TRIANGLES BECAUSE
C****THE NODE NUMBERS ARE DIFFERENT SO CONNECTIONS JOINING 1ST
C****NODE OF TARGET (OR ITS REFLECTION)TO NODE BELOW AND TO THE
C****RIGHT MUST BE ELIMINATED.
      IF(IABS(M1-M2) .EQ. 1 .AND. IABS(M1-M3) .EQ. NXNY
     *.AND. (M1 .EQ. ISTARG .OR. M1 .EQ. (ISTARG + NONODE)))GO TO 490
      IF(M1 .LE. NONODE)GO TO 410
      IT1 = KT1 - NONODE
      IT2 = KT2 - NONODE
      IT3 = KT3 - NONODE
C**** DO NOT PLOT TRIANGLE JOINING UPPER RIGHT FRONT CORNER OF
C**** FULL TARGET TO POINT BEHIND IT AND BELOW IT(NOTABLY FOR
C**** POSITIVE ROTATION FROM X TO Y AXIS
      IF(IABS(M1 - M2) .EQ. NX .AND. IABS(M2 - M3)
     * .EQ. NXNY
     * .AND. IT2 .EQ. NXNY)GO TO 490
C**** DO NOT PLOT TRIANGLE JOINING LOWER RIGHT FRONT CORNER
C****  OF TARGET TO POINT BEHIND IT AND POINT ON ITS LEFT
      IF(IABS(M2-M3) .EQ. 1 .AND. IABS(M1-M3) .EQ. NX .AND.
     * IT1 .GT. NXNY)GO TO 490
C**** UPPER LEFT FRONT CORNER
      IF(IABS(M1-M2) .EQ. NX .AND. IABS(M2-M3) .EQ. NXNY
     * .AND. M2 .EQ. NXNYMR)GO TO 490
        GO TO 410
C
C**** HALF TARGET
C
C**** UPPER LEFT CORNER
  405 IF(IABS(M1-M2) .EQ. NX .AND. IABS(M1-M3) .EQ. NXNY
     * .AND. MOD(KT1,NX) .EQ. 1)GO TO 490
C**** LOWER RIGHT CORNER
      IF(IABS(M1-M2) .EQ. 1 .AND. IABS(M2-M3) .EQ. NX
     * .AND. KT1 .GT. NXNY)GO TO 490
C**** UPPER RIGHT CORNER
      IF(IABS(M1-M2) .EQ. NX .AND. IABS(M1-M3) .EQ. NXNY
     * .AND. MOD(KT1,NX) .EQ. 0 .AND. MOD(KT2,NX) .EQ. C
     * .AND. MOD(KT3,NX) .EQ. 0)
     * GO TO 490
C**** LOWER LEFT CORNER
      IF(IABS(M1-M2) .EQ. 1 .AND. IABS(M1-M3) .EQ. NX
     * .AND. MOD(KT1,NX) .EQ. 1
     * .AND. KT1 .GT. NXNY)GO TO 490
      GO TO 410
C

C**** PROJECTILE ONLY - WITH REFLECTION
```

```
C
  407 IF(M1 .LE. IMOD .AND. (M2 .GT. NODLAY .OR. M3 .GT. NODLAY))
     *GO TO 490
      IF(M1 .GT. NONODE .AND. M1 .LE. JMOD .AND.
     *(M2 .GT. KMOD .OR. M3 .GT. KMOD))GO TO 490
  410     NP(1)=M1
          NP(2)=M2
          NP(3)=M3
          NP(4)=M1
  420         DO 430 I=1,NPNTS
              J=NP(I)
              XP(I)=TRANSF(J,1)
              ZP(I)=TRANSF(J,3)
  430         CONTINUE
C     WRITE(6,4)(XP(I),ZP(I),I=1,6)
C   4 FORMAT(12F9.3)
  431 IF(ITARGT .EQ. 0)GO TO 435
      CALL PLOT3D
      GO TO 490
C****IF NO TARGET, PLOT PROJECTILE WITHOUT SETTING UP
C****HIDDEN LINE MASK FOR TARGET.
  435 KK = 3
      DO 440 I = 1,NPNTS
      AX = (XP(I)-XP(5))/XP(6)
      AY = (ZP(I)-ZP(5))/ZP(6)
      CALL PLOT(AX,AY,KK)
      KK = 2
  440 CONTINUE
  490     CONTINUE
      DO 495 I = 1,LIMIT
  495 ITRIAN(I,1) = 0
      IF(ITARGT .NE. 1 .OR. IMAT .EQ. 2)GO TO 605
C**** INITIALIZE IN FOR CKTRI
      IN = 1
      IMAT = 2
      CALL CKMASK
C**** INITIALIZE IND1 AND IND2 FOR CKPTS
      IND1 = ISTARG
      IND2 = NONODE
      GO TO 305
  600 CONTINUE
      IF(IPLOT .EQ. 0)GO TO 200
  605 IF(IOUTPT .EQ. 1)GO TO 608
      CALL PLOT(0.0,0.0,999)
      IF(IOUTPT .EQ. 3)GO TO 183
      CALL HDCOPY
      CALL TINPUT(IDUM)
      GO TO 183
  608 OVER = OVER + 13.
      CALL PLOT(OVER,1.25,-3)
      GO TO 183
  610 REWIND 3
      CALL PLOT(0.,0.,999)
      STOP
  520 WRITE (6,180)
      STOP
      END
```

```
      SUBROUTINE TRANS
      INCLUDE COMDIM
   30 CALL IDENT
  200 READ(7,1)IWORD
    1 FORMAT(I1)
C**** SCALING
      IF(IWORD .EQ. 1) GO TO 300
C**** TRANSLATION
      IF(IWORD .EQ. 2) GO TO 400
C**** ROTATION
      IF(IWORD .EQ. 3) GO TO 500
C**** PLOT
      IF(IWORD .EQ. 4) RETURN
C**** START OVER
      IF(IWORD .EQ. 5) GO TO 30
      GO TO 200
  300 READ(7,2)IS1,IS2,IS3
    2 FORMAT(6I4)
      S1 = IS1/100.
      S2 = IS2/100.
      S3 = IS3/100.
      CALL SCALIN(S1,S2,S3)
      GO TO 200
  400 READ(7,2)IT1,IT2,IT3
      T1 = IT1/100.
      T2 = IT2/100.
      T3 = IT3/100.
      CALL TRANSL(T1,T2,T3)
      GO TO 200
  500 READ(7,2)IR1,IR2,ITHETA
  550 CALL ROTATE(IR1,IR2,ITHETA)
      GO TO 200
      END
```

```
      SUBROUTINE IDENT
C**** INITIALIZE TRANSFORMATION MATRICES
      INCLUDE COMDIM
      DO 10 I = 1,4
      DO 5 J = 1,4
      ARRINT(I,J) = 0
      ARRAY(I,J) = 0
    5 ARRCHG(I,J) = 0
   10 CONTINUE
      DO 20 I = 1,4
      ARRAY(I,I) = 1
   20 ARRINT(I,I) = 1
      RETURN
      END
```

```
      SUBROUTINE SCALIN(S1,S2,S3)
C**** PERFORM SCALING
      INCLUDE COMDIM
      ARRAY(1,1) = S1
      ARRAY(2,2) = S2
      ARRAY(3,3) = S3
      CALL MULMAT
      RETURN
      END
```

```
      SUBROUTINE TRANSL(T1,T2,T3)
C**** PERFORM TRANSLATION
      INCLUDE COMDIM
      ARRAY(4,1) = T1
      ARRAY(4,2) = T2
      ARRAY(4,3) = T3
      CALL MULMAT
      RETURN
      END
```

```
      SUBROUTINE ROTATE(N1,N2,ITHETA)
C**** PERFORM ROTATION OF MATRICES
      INCLUDE COMDIM
      THETA = ITHETA
      STHETA = SIN(THETA * .017453293)
      CTHETA = COS(THETA * .017453293)
      ARRAY(N1,N1) = CTHETA
      ARRAY(N2,N2) = CTHETA
      ARRAY(N1,N2) = STHETA
      ARRAY(N2,N1) = - STHETA
      CALL MULMAT
      RETURN
      END
```

```
      SUBROUTINE MULMAT
C**** MULTIPLY MATRICES
      INCLUDE COMDIM
      DO 10 I = 1,4
      DO 5 J = 1,4
      DO 3 K = 1,4
    3 ARRCHG(I,J) = ARRAY(I,K) * ARRINT(K,J) + ARRCHG(I,J)
    5 CONTINUE
   10 CONTINUE
      DO 30 I = 1,4
      DO 20 J = 1,4
      ARRINT(I,J) = ARRCHG(I,J)
      ARRAY(I,J) = 0
   20 ARRCHG(I,J) = 0
   30 CONTINUE
      DO 40 I = 1,4
   40 ARRAY(I,I) = 1
      RETURN
      END
```

```
      SUBROUTINE TOP(TIME,CYCLE,TITLE)
C        PLOT TITLE AND HEADER INFORMATION
C
         DIMENSION TITLE(3)
      CALL SYMBOL(2.5,8.,.14,TITLE,0.0,18)
      CALL SYMBOL(6.0,8.,.14,6HTIME =,0.0,6)
      CALL NUMBER(999.,999.,.14,TIME,0.0,8)
      RETURN
      END
```

```
      SUBROUTINE CKTRI (N1,N2,N3,N4)
C**** THIS ROUTINE CHECKS FOR INTERIOR TRIANGLES(SHOULD BE
C**** DUPLICATED) AND DROPS THEM.
      INCLUDE COMDIM,LIST
      DIMENSION NUM(4,2),NN(4)
      DATA MULT1/10000/
C
C                          *****FORMAT STATEMENTS*****
C
  110 FORMAT (' ITRIAN NEEDS TO BE REDIMENSIONED LARGER')
C
      IN = IN
      NN(1)=N1
      NN(2)=N2
      NN(3)=N3
      NN(4)=N4
      ICOUNT=ICOUNT+1
C**** SORT NODAL POINTS SO THAT N1<N2<N3<N4
      M=2
         DO 130 K=1,3
            DO 120 L=M,4
            IF (NN(L).GT.NN(K)) GO TO 120
            ISAVE=NN(K)
            NN(K)=NN(L)
            NN(L)=ISAVE
  120       CONTINUE
         M=M+1
  130    CONTINUE
C**** PACK THE NODAL NUMBERS OF THE THREE POINTS MAKING UP TRIANGLE
C****INTO 1 WORD IN ITRIAN. IF POINTS 1000,2000,3000 MAKE UP TRIANGLE,
C****ITRIAN(I,1) = 10002000 AND ITRIAN(I,2) = 3000
      NUM(1,1) = NN(1) * MULT1 + NN(2)
      NUM(1,2) = NN(3)
      NUM(2,1) = NUM(1,1)
      NUM(2,2) = NN(4)
      NUM(3,1) = NN(1) * MULT1 + NN(3)
      NUM(3,2) = NN(4)
      NUM(4,1) = NN(2) * MULT1 + NN(3)
      NUM(4,2) = NN(4)
C**** CHECK THIS TRIANGLE AGAINST OTHERS ALREADY PROCESSED. IF A
C**** TRIANGLE IS PROCESSED TWICE, IT MUST BE AN INTERIOR TRIANGLE,
C**** SO IT IS REMOVED FROM ITRIAN (ITRIAN(I,1)=0) AND NOT CHECKED
C**** AGAINST POINTS TO SEE IF IT HIDES THEM.
         DO 170 K=1,4
            DO 140 L=1,IN
      IF(ITRIAN(L,1) .NE. NUM(K,1))GO TO 140
      IF(ITRIAN(L,2) .EQ. NUM(K,2))GO TO 150
  140       CONTINUE
         ITRIAN(IN,1)=NUM(K,1)
      ITRIAN(IN,2) = NUM(K,2)
         IN=IN+1
         IF (IN.GT.LIMIT) GO TO 160
         GO TO 170
  150    ITRIAN(L,1)=0
         GO TO 170
```

```
C**** ITRIAN IS DIMENSIONED TO LIMIT. IF AN OVERFLOW OCCURS, GO
C**** BACK AND FILL IN SPACES FROM WHICH TRIANGLES WERE REMOVED.
 160      CALL PACK (INDEX)
          IN=INDEX
          IF (IN.GT.LIMIT) WRITE (6,110)
 170      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PACK(INDEX)
C**** THIS ROUTINE PACKS ITRIAN. BLANK SPOTS WERE CREATED WHEN DUPLICATE
C**** TRIANGLES WERE REMOVED.
      INCLUDE COMDIM,LIST
      M = LIMIT
      DO 10 J = 1,LIMIT
      IF(ITRIAN(J,1) .NE. 0)GO TO 10
      DO 5 I = M,1,-1
      IF(ITRIAN(I,1) .NE. 0)GO TO 8
    5 CONTINUE
      GO TO 20
    8 IF(J .GT. I)GO TO 20
      ITRIAN(J,1) = ITRIAN(I,1)
      ITRIAN(J,2) = ITRIAN(I,2)
      ITRIAN(I,1) = 0
      M = I - 1
      IF(J .GE. M)GO TO 15
   10 CONTINUE
   15 INDEX = J
      GO TO 25
   20 INDEX = J - 1
   25 WRITE(6,1)(ITRIAN(I,1),I=1,INDEX)
    1 FORMAT(2I20)
      RETURN
      END
```

```
      SUBROUTINE CKPTS(IND1,IND2)
C**** THIS ROUTINE CHECKS ALL POINTS DEFINING EXTERIOR TRIANGLES
C**** AGAINST ALL EXTERIOR TRIANGLES TO SEE WHICH POINTS ARE
C**** HIDDEN AND SHOULD NOT BE CONNECTED.
      INCLUDE COMDIM,LIST
C
C                          *****FORMAT STATEMENTS*****
C
 120  FORMAT (6F10.3,I2)
C
         DO 480 JJ=1,LIMIT
         IF (ITRIAN(JJ,1).EQ.0) GO TO 480
      M1 = ITRIAN(JJ,1)/10000
      M2 = ITRIAN(JJ,1) - (M1 * 10000)
      M3 = ITRIAN(JJ,2)
C****IF ALL 3 POINTS OF TRIANGLE HAVE SAME X OR Z COORDINATES,DO
C****NOT CONSIDER THIS TRIANGLE.
      IF(TRANSF(M1,1) .EQ. TRANSF(M2,1) .AND.
     *   TRANSF(M1,1) .EQ. TRANSF(M3,1))GO TO 480
      IF(TRANSF(M1,3) .EQ. TRANSF(M2,3) .AND.
     *   TRANSF(M1,3) .EQ. TRANSF(M3,3))GO TO 480
C****IF 2 POINTS OF TRIANGLE HAVE SAME Z AND X COORDINATES,
C****DO NOT CONSIDER THIS TRIANGLE.
         IF ((TRANSF(M1,3).EQ.TRANSF(M2,3).AND.TRANSF(M1,1)
     1   .EQ.TRANSF(M2,1)).OR.(TRANSF(M1,3).EQ.TRANSF(M3,3)
     2   .AND.TRANSF(M1,1).EQ.TRANSF(M3,1)).OR.(TRANSF(M2,3)
     3   .EQ.TRANSF(M3,3).AND.TRANSF(M2,1).EQ.TRANSF(M3,1))) GO TO 480
C****DETERMINE 1ST POINT OF TRIANGLE
C**** THIS IS ARBITRARILY ONE WITH LOWEST Z VALUE.
         IF (TRANSF(M1,3)-TRANSF(M2,3)) 150,140,130
 130     IF (TRANSF(M2,3)-TRANSF(M3,3)) 190,200,160
 140     IF (TRANSF(M1,3)-TRANSF(M3,3)) 170,480,160
 150     IF (TRANSF(M1,3)-TRANSF(M3,3)) 180,210,160
 160     IPT1=M3
         IPT2=M2
         IPT3=M1
         GO TO 220
 170     IF (TRANSF(M1,1)-TRANSF(M2,1)) 180,480,190
 180     IPT1=M1
         IPT2=M2
         IPT3=M3
         GO TO 220
 190     IPT1=M2
         IPT2=M3
         IPT3=M1
         GO TO 220
 200     IF (TRANSF(M2,1)-TRANSF(M3,1)) 190,480,160
 210     IF (TRANSF(M1,1)-TRANSF(M3,1)) 180,480,160
C**** DETERMINE SLOPES BETWEEN POINT 1 AND OTHER 2 POINTS
C****IN ORDER TO DETERMINE POINTS 2 AND 3 SO THAT 1-3
C****DEFINES TRIANGLE IN COUNTER CLOCKWISE ORDER.
C****DETERMINE SLOPE BETWEEN POINTS 1 AND 2.
 220     IF (TRANSF(IPT1,1)-TRANSF(IPT2,1)) 240,230,240
 230     AM1=99999.
         GO TO 250
```

```
 240      AM1=(TRANSF(IPT1,3)-TRANSF(IPT2,3))/(TRANSF(IPT1,1)-
     1    TRANSF(IPT2,1))
          IF (ABS(AM1).LT..0001) AM1=0.0
C****DETERMINE SLOPE BETWEEN POINTS 1 AND 3.
 250      IF (TRANSF(IPT1,1)-TRANSF(IPT3,1)) 270,260,270
 260      AM3=99999.
          GO TO 280
 270      AM3=(TRANSF(IPT1,3)-TRANSF(IPT3,3))/(TRANSF(IPT1,1)-
     1    TRANSF(IPT3,1))
          IF (ABS(AM3).LT..0001) AM3=0.0
C****IF BOTH SLOPES ARE POSITIVE OR BOTH ARE NEGATIVE,TAKE POINT
C****WHOSE SLOPE WITH POINT 1 IS SMALLER.
 280      IF ((AM1.GE.0..AND.AM3.GE.0.).OR.(AM1.LT.0.0.AND.AM3.LT.0.))
     1     GO TO 290
C****IF ONE SLOPE IS POSITIVE,THE OTHER NEGATIVE,POINT 2 IS ONE
C****WHOSE SLOPE WITH POINT1 IS POSITIVE.
          IF (AM1.GE.0.) GO TO 310
          GO TO 300
 290      IF (AM1.LT.AM3) GO TO 310
 300      SAVE=AM3
          ISAVE=IPT3
          AM3=AM1
          AM1=SAVE
          IPT3=IPT2
          IPT2=ISAVE
 310      CONTINUE
C****DETERMINE SLOPE BETWEEN POINTS 2 AND 3.
          IF (TRANSF(IPT2,1)-TRANSF(IPT3,1)) 330,320,330
 320      AM2=99999.
          GO TO 340
 330      AM2=(TRANSF(IPT2,3)-TRANSF(IPT3,3))/(TRANSF(IPT2,1)-
     1    TRANSF(IPT3,1))
          IF (ABS(AM2).LT..0001) AM2=0.0
C**** DETERMINE WHICH POINTS ARE HIDDEN BY THIS TRIANGLE.
C**** IF HIDDEN, INTEL(I) = 1
 340      LL=IND1
          MM=IND2
             DO 470 KK=1,2
             DO 460 I=LL,MM
C****IF POINT ALREADY HIDDEN BY A TRIANGLE DONT TEST IT AGAIN.
             IF (INTEL(I).EQ.1) GO TO 460
C****IF Y(POINT) <OR= Y1,Y2,Y3, IT CANT BE HIDDEN BY TRIANGLE
             IF (TRANSF(I,2).LE.TRANSF(IPT1,2).AND.TRANSF(I,2)
     1       .LE.TRANSF(IPT2,2).AND.TRANSF(I,2).LE.TRANSF(IPT3,2))
     2        GO TO 460
C****IF X(POINT)>OR= X1,X2,X3, IT CANT BE HIDDEN BY TRIANGLE
             IF (TRANSF(I,1).GE.TRANSF(IPT1,1).AND.TRANSF(I,1)
     1       .GE.TRANSF(IPT2,1).AND.TRANSF(I,1).GE.TRANSF(IPT3,1))
     2        GO TO 460
C****IF X(POINT<OR= X1,X2,X3, IT CANT BE HIDDEN BY TRIANGLE
             IF (TRANSF(I,1).LE.TRANSF(IPT1,1).AND.TRANSF(I,1)
     1       .LE.TRANSF(IPT2,1).AND.TRANSF(I,1).LE.TRANSF(IPT3,1))
     2        GO TO 460
C****IF Z(POINT)>OR= Z1,Z2,Z3, IT CANT BE HIDDEN BY TRIANGLE
             IF (TRANSF(I,3).GE.TRANSF(IPT1,3).AND.TRANSF(I,3)
```

```
     1          .GE.TRANSF(IPT2,3).AND.TRANSF(I,3).GE.TRANSF(IPT3,3))
     2           GO TO 460
C****IF Z(POINT)<OR= Z1,Z2,Z3, IT CANT BE HIDDEN BY TRIANGLE
                 IF (TRANSF(I,3).LE.TRANSF(IPT1,3).AND.TRANSF(I,3)
     1          .LE.TRANSF(IPT2,3).AND.TRANSF(I,3).LE.TRANSF(IPT3,3))
     2           GO TO 460
C****IF POINT = POINT1 OF TRIANGLE,IT CANT BE HIDDEN BY TRIANGLE
                IF (TRANSF(I,1).EQ.TRANSF(IPT1,1).AND.TRANSF(I,2)
     1          .EQ.TRANSF(IPT1,2).AND.TRANSF(I,3).EQ.TRANSF(IPT1,3))
     2           GO TO 460
C****IF POINT = POINT2 OF TRIANGLE, IT CANT BE HIDDEN BY TRIANGLE
                IF (TRANSF(I,1).EQ.TRANSF(IPT2,1).AND.TRANSF(I,2)
     1          .EQ.TRANSF(IPT2,2).AND.TRANSF(I,3).EQ.TRANSF(IPT2,3))
     2           GO TO 460
C****IF POINT = POINT3 OF TRIANGLE, IT CANT BE HIDDEN BY TRIANGLE
                IF (TRANSF(I,1).EQ.TRANSF(IPT3,1).AND.TRANSF(I,2)
     1          .EQ.TRANSF(IPT3,2).AND.TRANSF(I,3).EQ.TRANSF(IPT3,3))
     2           GO TO 460
                ISLOSW=0
C**** DETERMINE SLOPES BETWEEN POINT AND POINTS 1 AND 2 OF TRIANGLE
                IF (TRANSF(I,1)-TRANSF(IPT1,1)) 360,350,360
 350            AM1P=99999.
                GO TO 370
 360            AM1P=(TRANSF(I,3)-TRANSF(IPT1,3))/(TRANSF(I,1)-
     1          TRANSF(IPT1,1))
                IF (ABS(AM1P).LT..0001) AM1P=0.0
 370            IF (TRANSF(I,1)-TRANSF(IPT2,1)) 390,380,390
 380            AM2P=99999.
                GO TO 400
 390            AM2P=(TRANSF(I,3)-TRANSF(IPT2,3))/(TRANSF(I,1)-
     1          TRANSF(IPT2,1))
                IF (ABS(AM2P).LT..0001) AM2P=0.0
 400            IF (AM1P.GE.AM1.AND.AM1P.LE.AM3) ISLOSW=ISLOSW+1
                IF (AM2P.LE.AM1.AND.AM2P.GE.AM2) ISLOSW=ISLOSW+1
                IF (ISLOSW.EQ.2) GO TO 440
                IF (TRANSF(I,1)-TRANSF(IPT3,1)) 420,410,420
 410            AM3P=99999.
                GO TO 430
 420            AM3P=(TRANSF(I,3)-TRANSF(IPT3,3))/(TRANSF(I,1)-
     1          TRANSF(IPT3,1))
                IF (ABS(AM3P).LT..0001) AM3P=0.0
 430            IF (AM3P.LE.AM2.AND.AM3P.GE.AM3) ISLOSW=ISLOSW+1
                IF (ISLOSW.NE.2) GO TO 460
 440            IF (TRANSF(I,2).GE.TRANSF(IPT1,2).AND.TRANSF(I,2)
     1          .GE.TRANSF(IPT2,2).AND.TRANSF(I,2).GE.TRANSF(IPT3,2))
     2           GO TO 450
C**** DETERMINE EQUATION OF PLANE CONTAINING TRIANGLE
                A=TRANSF(IPT1,2)*(TRANSF(IPT2,3)-TRANSF(IPT3,3))-
     1          TRANSF(IPT1,3)*(TRANSF(IPT2,2)-TRANSF(IPT3,2))+
     2          (TRANSF(IPT2,2)*TRANSF(IPT3,3))-(TRANSF(IPT3,2)*
     3          TRANSF(IPT2,3))
                B=-(TRANSF(IPT1,1)*(TRANSF(IPT2,3)-TRANSF(IPT3,3))-
     1          TRANSF(IPT1,3)*(TRANSF(IPT2,1)-TRANSF(IPT3,1))+
     2          TRANSF(IPT2,1)*TRANSF(IPT3,3)-TRANSF(IPT3,1)*TRANSF(IPT2,
     3          3))
```

```
                C=TRANSF(IPT1,1)*(TRANSF(IPT2,2)-TRANSF(IPT3,2))-
     1          TRANSF(IPT1,2)*(TRANSF(IPT2,1)-TRANSF(IPT3,1))+
     2          TRANSF(IPT2,1)*TRANSF(IPT3,2)-TRANSF(IPT3,1)*TRANSF(IPT2,
     3          2)
                D=-(TRANSF(IPT1,1)*(TRANSF(IPT2,2)*TRANSF(IPT3,3)-
     1          TRANSF(IPT3,2)*TRANSF(IPT2,3))-TRANSF(IPT1,2)*
     2          (TRANSF(IPT2,1)*TRANSF(IPT3,3)-TRANSF(IPT3,1)*
     3          TRANSF(IPT2,3))+TRANSF(IPT1,3)*(TRANSF(IPT2,1)*
     4          TRANSF(IPT3,2)-TRANSF(IPT3,1)*TRANSF(IPT2,2)))
                YPLANE=-(A*TRANSF(I,1)+C*TRANSF(I,3)+D)/B
C               WRITE (6,120) A,B,C,D,YPLANE,TRANSF(I,2),INTEL(I)
                IF (YPLANE.GE.TRANSF(I,2)) GO TO 460
 450            INTEL(I)=1
 460            CONTINUE
              IF (IR.EQ.0) GO TO 480
C**** CHECK POINTS ON REFLECTIVE TRIANGLES ALSO.
              LL=NONODE+ LL
              MM=MM+NONODE
 470          CONTINUE
 480       CONTINUE
           RETURN
           END
```

```
      SUBROUTINE PLOT3D
C**** THIS ROUTINE USES A MASKING TECHNIQUE WHEREIN THE UPPER AND LOWER Z'S
C**** FOR A GIVEN X ON THE PROJECTILE ARE STORED.  LATER, WHEN THE TARGET IS
C**** PLOTTED, LINES OR PORTIONS THEREOF ARE PLOTTED
C**** WHICH ARE ABOVE OR BELOW THIS MASK BUT NOT WITHIN IT.
      INCLUDE COMDIM
      INTEGER HIGH
      DATA IFIRST/0/
C
C                          *****FORMAT STATEMENTS*****
C
  110 FORMAT(' LOW = ',I6,'IN ST<ATEMENT IADD = 10000 - ICHECK,
     *CHANGE 10000 TO 0 IF LOW > 20000, OR 20000 IF LOW < 2')
  120  FORMAT (' ERR-LOW MASK > HIGH MASK',3I6)
C
         DO 670 K=1,NPNTS
      IF(K .EQ. 1)LOCSW = 0
         IX=(XP(K)*1000.)
         IY=(ZP(K)*1000.)
         IF (K.NE.1.AND.IMAT.EQ.1) GO TO 280
C     (LOC) IS THE POSITION OF THE PREVIOUS POINT WITH RESPECT
C     TO THE MASK
C     +1 ABOVE THE MASK
C     0 WITHIN THE LIMITS OF THE MASK
C     -1 BELOW THE MASK
C     PROCEDURE FOR INITIAL POINT OF EACH LINE
C     LOCATE INITIAL POINT WITH RESPECT TO THE MASK THEN
C     UPDATE THE MASK
         IMSKX=IX*SIN(ALPHA)+IY*COS(ALPHA)
      IF(IFIRST .EQ. 1)GO TO 125
      ICHECK = IMSKX + IMSKX
      IADD = 10000 - ICHECK
      WRITE(6,888)ICHECK,IADD
  888 FORMAT(2I5)
      IFIRST = 1
  125    LOW=IMSKX + IMSKX + IADD
         IF ((LOW.GT.20000.OR.LOW.LT.2).AND.IMAT.EQ.1) WRITE (6,110) LOW
C     WRITE(6,178)IX,IY,IMSKX,LOW
         HIGH=LOW-1
         IF ((HIGH.GE.ISTART.AND.HIGH.LE.IEND).OR.IMAT.EQ.1) GO TO 130
         MLOW=-10000
         MHIGH=-10000
      IF(HIGH .LT. ISTART)LOCSW = -1
      IF(HIGH .GT. IEND)LOCSW = 1
      IF(K .EQ. 1)LOCOSW = LOCSW
      IF(K .NE. 1 .AND. LOCOLD .NE. 0 .AND. LOCSW .EQ. LOCOSW)
     *GO TO 150
         GO TO 140
  130    MLOW=MASK(LOW)
         MHIGH=MASK(HIGH)
  140 CONTINUE
         IF (K.NE.1.AND.IMAT.EQ.2) GO TO 280
         IF (MHIGH-IY) 180,180,160
  150    CALL IPLOT (IX,IY,2)
         LOCOLD=+1
```

```
            GO TO 220
  160       IF (MLOW-IY) 170,200,200
  170       LOCOLD=0
            GO TO 210
  180       CONTINUE
            IF (IMAT.EQ.1) MASK(HIGH)=IY
            IF (MLOW.EQ.-10000.AND.IMAT.EQ.1) MASK(LOW)=IY
            LOCOLD=+1
            GO TO 210
  190       WRITE (6,120) MHIGH,JY,MLOW
            STOP
  200       CONTINUE
            IF (IMAT.EQ.1) MASK(LOW)=IY
            LOCOLD=-1
C       MOVE THE RAISED PEN TO THIS INITIAL POINT
  210       IF (LOCOLD.NE.0.OR.IMAT.EQ.1) CALL IPLOT (IX,IY,3)
  220       JX=IX
            JY=IY
            KX=IX
            KY=IY
            GO TO 670
C       SPECIAL CASE WHERE CHANGE IN X COORDINATE IS ZERO
C       A SPECIAL PROVISION IS MADE AT THIS POINT SO THAT A LINE
C       WILL NOT MASK ITSELF AS LONG AS THE X COORDINATE REMAINS
C       CONSTANT
  280       IF (IX.NE.JX) GO TO 290
            JY=IY
            GO TO 310
C       COMPUTE CONSTANTS FOR LINEAR INTERPOLATION
  290       YINC=FLOAT(IY-JY)/ABS(FLOAT(IX-JX))
            INCX=(IX-JX)/IABS(IX-JX)
            YJ=JY
C       PERFORM  LINEAR INTERPOLATION AT EACH INCREMENTAL STEP ON
C       THE X AXIS
  300       JX=JX+INCX
            YJ=YJ+YINC
            JY=YJ+.5
C       LOCATE THE CURRENT POINT WITH RESPECT TO THE MASK AT THAT
C       POINT THEN PLOT THE INCREMENT AS A FUNCTION OF THE
C       LOCATION OF THE PREVIOUS POINT WITH RESPECT TO ITS MASK
  310       CONTINUE
            IMSKX=JX*SIN(ALPHA)+JY*COS(ALPHA)
            LOW=IMSKX + IMSKX + IADD
            IF ((LOW.GT.20000.OR.LOW.LT.2).AND.IMAT.EQ.1) WRITE (6,110) LOW
            HIGH=LOW-1
            IF ((HIGH.GE.ISTART.AND.HIGH.LE.IEND).OR.IMAT.EQ.1) GO TO 320
            MLOW=-10000
            MHIGH=-10000
          IF(HIGH .LT. ISTART)LOCSW = -1
          IF(HIGH .GT. IEND)LOCSW = 1
            GO TO 330
  320       MLOW=MASK(LOW)
            MHIGH=MASK(HIGH)
  330       IF (MHIGH-JY) 360,350,340
  340       IF (MLOW-JY) 370,390,390
```

```
 350      IF (MLOW-JY) 360,380,190
C      THE CURRENT POINT IS ABOVE THE MASK
 360      LOC=+1
       IF(IMAT .EQ. 1)MASK(HIGH) = JY
       IF(IMAT .EQ. 1 .AND. MLOW .EQ. -10000)MASK(LOW) = JY
       IF(IMAT .EQ. 1)GO TO 650
          IF (LOCSW.NE.0.AND.LOCOLD.NE.0) GO TO 640
          IF (LOCOLD) 470,450,590
C      THE CURRENT POINT IS WITHIN THE MASK
 370      LOC=0
       LOCSW = 0
       IF(IMAT .EQ. 1)GO TO 650
          IF (LOCOLD) 470,650,470
 380      IF (LOCOLD) 390,370,360
C      THE CURRENT POINT IS BELOW THE MASK
 390      LOC=-1
       IF(IMAT .EQ. 1)MASK(LOW) = JY
       IF(IMAT .EQ. 1)GO TO 650
          IF (LOCSW.NE.0.AND.LOCOLD.NE.0) GO TO 640
          IF (LOCOLD) 640,460,470
 400      IF (LOCOLD) 420,650,410
C**** PLOT FROM ABOVE MASK TO TOP EDGE OF MASK
 410      IARG1=HIGHY
          IARG2=MHIGH
          GO TO 660
C**** PLOT FROM BELOW MASK TO BOTTOM EDGE OF MASK
 420      IARG1=LOWY
          IARG2=MLOW
          GO TO 660
 430      IF (ABS(KY-IARG1).GT.ABS(JY-IARG2)) GO TO 440
          IF (IARG1.NE.-10000) CALL IPLOT (KX,IARG1,2)
          GO TO 670
 440      IF (IARG2.NE.-10000) CALL IPLOT (JX,IARG2,2)
          GO TO 670
 450      CALL IPLOT (JX,JY,3)
          GO TO 590
 460      CALL IPLOT (JX,JY,3)
          GO TO 640
 470      IF (LOCOLD.NE.0.OR.IMAT.EQ.1) CALL IPLOT (KX,KY,2)
          IMSKX=KX*SIN(ALPHA)+KY*COS(ALPHA)
          MSKL=IMSKX + IMSKX + IADD
          IF ((MSKL.GT.20000.OR.MSKL.LT.2).AND.IMAT.EQ.1) WRITE (6,110)
     1     MSKL
          MSKH=MSKL-1
          IF ((MSKH.GE.ISTART.AND.MSKH.LE.IEND).OR.IMAT.EQ.1) GO TO 480
          LOWY=-10000
          HIGHY=-10000
       IF(MSKH .LT. ISTART)LOCSW = -1
       IF(MSKH .GT. IEND)LOCSW = 1
          GO TO 490
 480      LOWY=MASK(MSKL)
          HIGHY=MASK(MSKH)
 490      IF (LOC) 500,400,510
 500      CONTINUE
          GO TO 520
```

```
 510      CONTINUE
          IF (IMAT.EQ.1) MASK(MSKL)=KY
 520      IF (LOCOLD) 540,530,550
 530      IF (LOC) 610,650,600
C      PLOT FROM BELOW MASK TO ABOVE MASK
 540      IF (LOCSW.NE.0) GO TO 640
          IARG1=LOWY
          IARG2=MHIGH
          IARG3=HIGHY
          IARG4=MLOW
          ASSIGN 590 TO N
          GO TO 560
C      PLOT FROM ABOVE MASK TO BELOW MASK
 550      IF (LOCSW.NE.0) GO TO 640
          IARG1=HIGHY
          IARG2=MLOW
          IARG3=LOWY
          IARG4=MHIGH
          ASSIGN 640 TO N
 560      IF (ABS(KY-IARG1).GT.ABS(JY-IARG2)) GO TO 580
          IF (IARG1.NE.-10000) CALL IPLOT (KX,IARG1,2)
C      GET TO OTHER SIDE OF MASK
          IF (IARG3.NE.-10000) CALL IPLOT (KX,IARG3,3)
 570      CALL IPLOT (JX,JY,2)
          GO TO N
 580      IF (IARG4.NE.-10000) CALL IPLOT (JX,IARG4,2)
C**** GET TO OTHER SIDE OF MASK
          IF (IARG2.NE.-10000) CALL IPLOT (JX,IARG2,3)
          GO TO 570
C      PLOT FROM ABOVE THE MASK TO ABOVE THE MASK
 590      CONTINUE
          GO TO 650
C      PLOT FROM INSIDE MASK TO ABOVE MASK
 600      IARG1=HIGHY
          IARG2=MHIGH
          ASSIGN 590 TO N
          GO TO 620
C      PLOT FROM INSIDE MASK TO BELOW MASK
 610      IARG1=LOWY
          IARG2=MLOW
          ASSIGN 640 TO N
 620      IF (ABS(KY-IARG1).GT.ABS(JY-IARG2)) GO TO 630
          IF (IARG1.NE.-10000) CALL IPLOT (KX,IARG1,3)
          GO TO 570
 630      IF (IARG2.NE.-10000) CALL IPLOT (JX,IARG2,3)
          GO TO 570
C      PLOT FROM BELOW THE MASK TO BELOW THE MASK
 640      CONTINUE
          LOCSW=0
 650      KX=JX
          KY=JY
 660      LOCOLD=LOC
          IF (JX.NE.IX) GO TO 300
          IF (LOC.NE.0.OR.IMAT.EQ.1) CALL IPLOT (JX,JY,2)
          IF (LOCOLD.EQ.0.AND.LOC.EQ.0) GO TO 670
          IF (LOC.EQ.0) GO TO 430
 670      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE IPLOT(IX,IY,IP)
      INCLUDE COMDIM
    2 AX = IX/1000.
      AY = IY/1000.
      AX = (AX - XP(5))/XP(6)
      AY = (AY - ZP(5))/ZP(6)
      CALL PLOT(AX,AY,IP)
      RETURN
      END
```

```
      SUBROUTINE CKMASK
C**** THIS ROUTINE CHECKS MASK LIMITS TO SEE IF THEY FIT WITHIN RANGE 1-2000(
      INCLUDE COMDIM,LIST
C
C                              *****FORMAT STATEMENTS*****
C
 110  FORMAT (' MASK COMPLETELY EMPTY')
 120  FORMAT (2I6,F10.3)
C
         DO 130 I=1,20000,2
         IF (MASK(I).NE.-10000) GO TO 140
 130     CONTINUE
      WRITE (6,110)
      ISTART=30000
      IEND=30000
      GO TO 170
 140  ISTART=I
         DO 150 I=ISTART,20000,2
         IF (MASK(I).EQ.-10000) GO TO 160
 150     CONTINUE
 160  IEND=I-2
      WRITE (6,120) ISTART,IEND,ALPHA
 170  CONTINUE
      RETURN
      END
```

APPENDIX E

RETRIEVAL RUNSTREAM

```
@ASG,A RDUM22.
@USE 22,RDUM22.
@MAP ,TPF$.ABS
IN RLIB.RETREV
LIB RLIB.
LIB SEAPPD*TEKLIB2.
@XQT
```

APPENDIX F

PROCEDURE TO DETERMINE IF A POINT IS HIDDEN BY A TRIANGLE

The following procedure is used in the program to determine whether a point (P) is hidden by a triangle defined by points A, B, C.

1. P is not hidden by ΔABC if one of the following conditions is met:

   a. $x_P \geq \max (x_A, x_B, x_C)$

   b. $x_P \leq \min (x_A, x_B, x_C)$

   c. $z_P \geq \max (z_A, z_B, z_C)$

   d. $z_P \leq \min (z_A, z_B, z_C)$

   e. $y_P \leq \min (y_A, y_B, y_C)$

2. Arrange the points defining the triangle in counterclockwise order (P1, P2, P3) such that $z_1 = \min (z_A, z_B, z_C)$. (If $z_m = z_n = z_1$ then P1 is point wherein $x_1 = \min (x_m, x_n)$. Let the slope between any two points A and B be M(A, B). Then if 2 of the following 3 conditions are met the point (P) lies within the area of the triangle and must be further checked (see Step 3). Otherwise it is not hidden by ΔABC.

   a. $M(1,P) \geq M(1,2)$ and $M(1,P) \leq M(1,3)$

   b. $M(2,P) \leq M(1,2)$ and $M(2,P) \geq M(2,3)$

   c. $M(3,P) \leq M(2,3)$ and $M(3,P) \geq M(1,3)$.

3. Determine the equation of the plane defined by P1, P2, P3. If y_{plane} (at x_P, z_P) $< y_P$ then the point (P) is hidden by the triangle.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST, N. Klein 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E P. O. Box 209 St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Commander US Army Communications Rsch and Development Command ATTN: DRDCO-SGS Fort Monmouth, NJ 07703
1	Commander US Army Missile Research and Development Command ATTN: DRDMI-R Redstone Arsenal, AL 35809
1	Commander US Army Missile Materiel Readiness Command ATTN: DRSMI-AOM Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Research & Development Cmd ATTN: DRDTA-UL Warren, MI 48090
2	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) Dover, NJ 07801
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
5	Commander US Army Materials and Mechanics Research Center ATTN: DRXMR-T, Mr. J. Bluhm DRXMR-T, Dr. D. Roylance DRXMR-T, Dr. A.F. Wilde Dr. J. Mescall DRXMR-ATL Watertown, MA 02172
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army BMD Advanced Technology Center ATTN: Mr. P. Boyd, BMDATC-M P. O. Box 1500 Huntsville, AL 35807
3	Commander Naval Weapons Center ATTN: Code 4057 Code 4011, E.Lundstrom Code 3813, M. Backman China Lake, CA 93555
2	ADTC/DLJW (MAJ D. Matuska, LTC J. Osborn) Eglin AFB, FL 32542
3	Honeywell, Inc. Government & Aerospace Products Division ATTN: Mr. J. Blackburn Dr. G. Johnson Mr. R. Simpson 600 Second Street, NE Hopkins, MN 55343
1	Kaman Sciences Corporation ATTN: Dr. P. Snow 1500 Garden of the Gods Road Colorado Springs, CO 80933
3	Sandia Laboratories ATTN: Dr. W. Herrmann Dr. L. Bertholf Dr. J. W. Nunziato Albuquerque, NM 87115
1	Drexel University Dept of Mechanical Engineering ATTN: Dr. P. C. Chou 32nd and Chestnut Streets Philadelphia, PA 19104
3	Stanford Research Institute ATTN: Dr. L. Seaman Dr. D. Curran Dr. D. Shockey 333 Ravenswood Avenue Menlo Park, CA 94025
	Aberdeen Proving Ground Dir, USAMSAA Cdr, USATECOM ATTN: DRSTE-SG-H